3-21-2013

Development of Autonomous Optimal Cooperative Control in Relay Rover Configured Small Unmanned Aerial Systems

Timothy J. Shuck

DEVELOPMENT OF AUTONOMOUS OPTIMAL COOPERATIVE CONTROL IN RELAY ROVER CONFIGURED SMALL UNMANNED AERIAL SYSTEMS

THESIS

Timothy J. Shuck
1st Lieutenant, USAF

AFIT-ENV-13-M-27

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENV-13-M-27

DEVELOPMENT OF AUTONOMOUS OPTIMAL COOPERATIVE CONTROL IN RELAY ROVER CONFIGURED SMALL UNMANNED AERIAL SYSTEMS

THESIS

Presented to the Faculty

Department of Systems and Engineering Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

TIMOTHY J. SHUCK, USAF
1st Lieutenant, USAF

March 2013

AFIT-ENV-13-M-27

DEVELOPMENT OF AUTONOMOUS OPTIMAL COOPERATIVE CONTROL IN RELAY ROVER CONFIGURED SMALL UNMANNED AERIAL SYSTEMS

Timothy J. Shuck, B.S.

1st Lieutenant, USAF

Approved:

Dr. David R. Jacques (Chairman)	Date
Dr. John M. Colombi (Member)	Date
Dr. Richard G. Cobb (Member)	Date

AFIT-ENV-13-M-27

Abstract

This thesis documents the research effort to develop, integrate and implement the system hardware and the software necessary to validate the Air Force Institute of Technology's theoretical advances in small unmanned aerial systems (SUAS) cooperative control. The end state objective of the research effort was to flight test an autonomous control algorithm on a communication relay unmanned aerial vehicle (UAV) that was actively relaying data to and from a rover UAV. The relay UAV is one part of a SUAS designed to utilize cooperative control to extend the effective line-of-sight operating range for a rover UAV.

An algorithm is integrated into ground control software that takes telemetry data (the current position of the ground station, rover UAV, and relay UAV) to determine where to navigate the relay aircraft for optimal communication signal strength. The ground station operator flies the rover aircraft in the extended line-of-sight operational envelope just as she/he would in the normal line-of-sight operations. The relay UAV is autonomously routed to the optimal communications relay position.

The research yielded a SUAS based on the Ardupilot Mega 2.0. Flight testing demonstrated the SUAS's ability to generate the correct navigation data autonomously; however, the navigation data was not successfully activated as current waypoints on the relay UAV's autopilot. Software in the loop testing was utilized to verify a solution to activate the navigation data but flight testing was not conducted to verify the simulation results.

To my family for supporting me and my country for being worth fighting for!

Acknowledgments

It has taken a team effort to make this research effort successful. Dr. Jacques has been critically important in providing guidance and resources to make this research effort possible. Rick Patton and Don Smith provided essential technical support and flight testing support. Mark Smearcheck provided programming expertise that allowed the research objectives to be realized, going well beyond the original scope of effort presented to him. Appreciation has been earned by the prior AFIT UAV research team members. It is their research on which this research built. Charles Neal came in at a critical time and provided a wealth of insight about the autopilot that made flight testing much more successful. Additionally, I want to thank Jon Welborn and Scott Songer. You have been great team mates throughout the research effort.

In addition to the people providing direct support to the research, I would like to recognize the loved ones in my life that have sustained me and made sacrifices to help me reach my goals. My wife has been amazing in her capacity to love and support me, thank you. To my son, thank you for loving daddy despite that he has to spend lots of time at work. My parents have made a life time of sacrifices to enable me to achieve challenging things. You are both amazing. For all the words of encouragement and unconditional love I am forever grateful. Finally to the United States Air Force, my choice to serve has opened up a world of opportunities I did not even dream of. Thank you.

Timothy J. Shuck, 1st Lieutenant, USAF

Table of Contents

List of Figures

List of Tables

Page

DEVELOPMENT OF AUTONOMOUS OPTIMAL COOPERATIVE CONTROL IN RELAY ROVER CONFIGURED SMALL UNMANNED AERIAL SYSTEMS

I. Introduction

1.1 Background

Current military utilization of unmanned aerial systems is extensive, with over 500,000 flight hours in 2010 and the Pentagon's spending on unmanned aerial systems is projected to be nearly four billion United States dollars annually [1]. In 2009 the United States Air Force published an Unmanned Aircraft Systems Flight Plan that identified small unmanned aerial systems (SUAS) as "a profound technological advance in air warfare by providing…life-saving situational awareness." The flight plan also identified the need to advance cooperative interaction of SUAS to extend the effective line-of-sight operational range [2]. There have been many research efforts into SUAS cooperative control configurations; however, flight testing to verify the theoretical advances has been limited [3]. The Air Force Institute of Technology (AFIT) has been actively pursuing flight testing of cooperative control in SUAS since 2008 [4].

An AFIT SUAS cooperative control research effort has been targeted at extending the line-of-sight operational range for SUAS. The objective is to use autonomous vehicles relaying communication signals to extend the operational range for a more distant unmanned aerial vehicle (UAV), known as a rover, with the relay UAV operating in an autonomous manner. This objective required advances in automation and cooperative control of SUAS. Optimal control is the approach that AFIT researchers adopted to solve the relay placement portion of the cooperative control research objectives. The optimal control approach required identifying not only the theoretical solution but also an implementable real-time algorithm. The optimal control theory and a proposed implementation are detailed in the article *Optimal Guidance of a Relay Aircraft to Extend Small Unmanned Aircraft Range* [4]. The automation advances required to meet the objective are detailed in Boire [5].

1.2 Problem Statement

This research effort builds on the advances AFIT's SUAS cooperative control researchers have developed since 2008. Development, integration and implementation of the system hardware and the software necessary to validate AFIT's theoretical advances in SUAS cooperative control was conducted. The end state objective of the research effort was to flight test an autonomous control algorithm on a relay UAV that was actively relaying data to a rover UAV in an extended effective line-of-sight operating range. As can be seen in Figure 1, the relay UAV completes the data link from the ground station to the rover UAV and back from the rover UAV to the ground station.

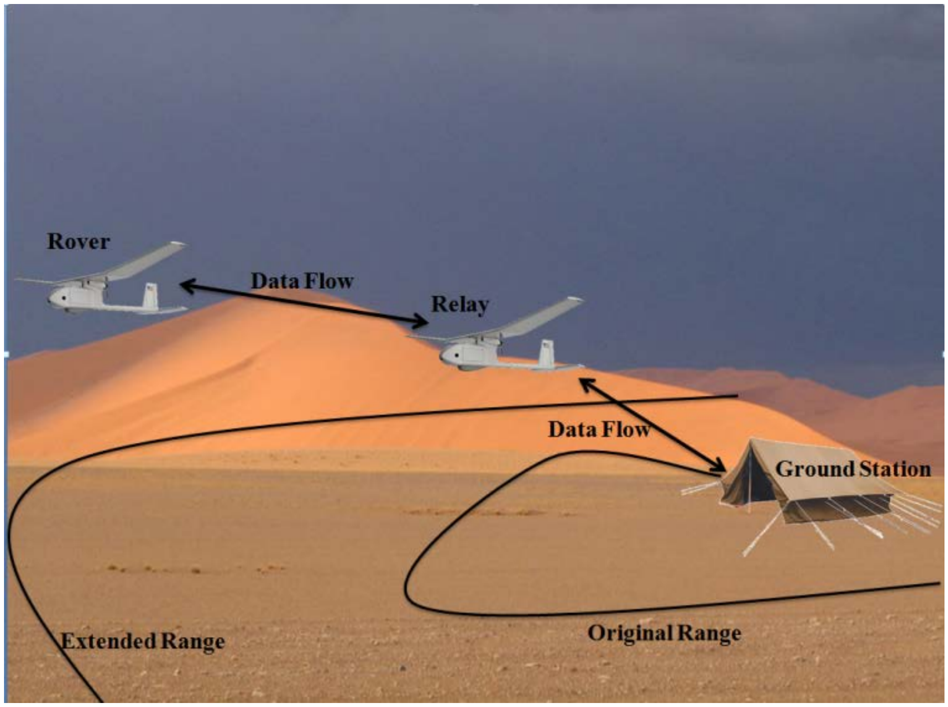

Figure 1. Simplified Operational View One (OV-1)

1.3 Scope

This thesis is one part of a larger research effort to develop cooperative control in SUAS. Advances in cooperative control theory and calculations for optimal control of aircraft trajectories theory are not redeveloped but are instead referenced [4] [5] [6] [7]. The focus of this thesis is development, integration, implementation, and testing for a cooperative control rover relay SUAS. The theory will either be validated or refuted by the test data.

System development, integration and implementation included: requirements analysis, system architecture analysis, selecting hardware (airframe, autopilot, sensors, communication and control), selecting ground control software, modifying hardware, modifying software, and finally integrating the system. A combination of government-off-the-shelf (GOTS) and commercial-off-the-shelf (COTS) hardware and open source software were utilized.

1.4 Methodology

The methodology applied to this research effort followed the “Vee” process model as described by Forsberg, Mooz and Cotterman [8]. The use of GOTS and COTS components accelerated some phases of the process but simultaneously lengthened other phases. Testing was integral to the research effort as it identified capability gaps and triggered iterative “Vee” cycles inside the larger “Vee” process.

1.5 Document Outline

Chapter I describes the introduction, problem statement, scope and general methodology of this thesis. Chapter II is a literature review of the current body of knowledge on SUAS cooperative control. Emphasis was placed on information that applied to the development, integration and implementation of the system hardware and the software necessary to validate AFIT’s theoretical advances in SUAS cooperative control. Chapter III describes the methodology. The methodology steps through the “Vee” process model and identifies key design decisions and the analysis process used to determine those design decisions. Chapter IV describes the degree of success produced by the methodology. Finally, Chapter V describes conclusions of the research effort and recommendations for further research.

II. Literature Review

2.1 Introduction

Many documents have been written that lay the foundation to enable a rover relay cooperative control configuration in field testing. This chapter will identify key documents that were influential in configuration choices and motivate the research subject. Additionally, this chapter will identify key foundational documents that have led up to the rover relay cooperative control configuration being developed to the point of enabling field testing. Finally, conclusions from the literature review will be discussed.

2.2 Supporting Research

Ryan et al. were commissioned by the Office of Naval Research to conduct a survey of recent research on the topic of cooperative control of UAVs [3]. Specifically, the authors identified five major areas of active research in cooperative control with UAVs, namely aerial surveillance and tracking, collision and obstacle avoidance, formation reconfiguration, high level control, and hardware/communications. AFIT's research in autonomous relay cooperative control most closely fits into Ryan et al.'s categories of high level control and hardware/communications. The most pertinent comment in the article relative to the present work was:

> "A major un-resolved issue for collaborative unmanned aircraft is wireless communication with other cooperating aircraft. The aircraft to ground problem generally involves out of line-of-sight, long range communications" [3, p. 603].

The authors' comment is of particular importance because they identify that no research has been completed that demonstrated a field-tested COTS solution to the wireless communication among cooperating UAVs [3]. This observation validates the need for the specific research objective this thesis addresses.

Fulghum and Dickerson examined the United States and international demand for unmanned aerial systems (UAS). They noted a growth in United States spending on UAS from $400 million in 1991 to nearly $4 billion in 2012. Flight hours of UAS have grown from 1,000 hours in 1987 to 500,000 hours in 2010. The authors project that Western countries' military demand for UAS will begin to slow through 2020; however, the Asian market for UAS technology will continue to increase as Asian countries catch up in UAS technology. This article supports continued research in UAS technology by identifying the growth and sustainability that the UAS market has demonstrated [1].

Air Force Doctrine Document 1 was created as "the Air Force's premier statement of our beliefs" [9, p. 3]. In this report the Air Force states that Intelligence, Surveillance, and Reconnaissance (ISR), provided by all UAS, is a foundational element of Air Force doctrine. The increased situational awareness gained by units using the cooperative control technology field-tested for this thesis will increase the unit's ability to seize, retain, and exploit the initiative. Understanding Air Force strategic doctrine influenced this research effort by providing context for potential future applications of the demonstrated technology. One example of this influence is the need to make the relay UAV fly autonomously to reduce operator load, thereby increasing the operator's situational awareness [9].

The Department of Defense has identified that reconnaissance and surveillance are the number one priority for combatant commanders when utilizing unmanned systems. Additionally, the Department of Defense identifies that full motion video is the most in-demand form of reconnaissance and surveillance. The primary research vehicle that has been selected for this thesis is the AFIT Overhead Watch and Loiter (OWL) aircraft. The OWL is a modified version of the RQ-11 aircraft that was originally designed, and is still field-deployed, to provide full motion video reconnaissance and surveillance. In the modification process to accommodate our research objectives, the full motion video capabilities of the aircraft were preserved. The relay aircraft must be able to relay not only the control signal to the rover, but full motion video signal from the rover to the ground station as well [10].

2.3 Foundational Research

Since 2008 AFIT has researched cooperative control to extend the range of SUAS. This section will step through key highlights of research work of the AFIT SUAS research team. The highlights are not intended to be all-inclusive of the body of knowledge leading up to development of a flight testable system but instead to provide background and a foundation for this thesis. For a more thorough examination of the research leading up to rover relay configured cooperative control field testing, the reader is directed to the foundational sources [4] [5] [6] [7].

Pachter, Hansen, Jacques, and Blue conducted research in 2008 intended, in their own words, to “develop guidance laws to optimally and autonomously position a relay Micro Aerial Vehicle (MAV) to provide an operator with real-time ISR by relaying communication and video signals from a rover MAV to the base, thus extending the rover's reach." Patcher et al. undertook the task of applying the approach of optimal control to solve the cooperative control problem. The objective of the optimal control problem was to position the communication node, in this case the relay UAV, to minimize the energy cost of communicating between a source and destination. In that process Patcher et al. developed the mathematical model that the AFIT SUAS research team would follow—up to and including the model used for this thesis. The model (Figure 2) simplified the analysis by reducing the three body problem to a planar scenario [4].

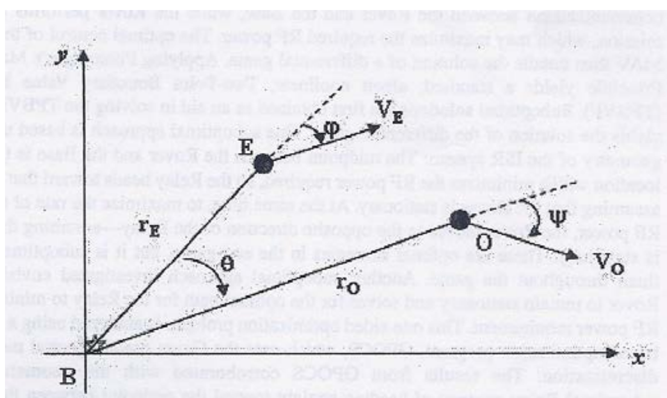

Figure 2. Schematic of Rover Relay System [4, p. 159]

For Figure 2 the following nomenclature was utilized:

B = Base
E = Relay SUAV
O = Rover SUAV
r_E = Distance from Base to Relay SUAV
r_O = Distance from base to Rover SUAV
V_E = Velocity of Relay
V_O = Velocity of Rover
Ψ = Relative Course Angle of Relay
Θ = Included Angle of the Radials from the Base to the Relay and the Rover
ϕ = Relative Course Angle of Rover [4].

Pachter et al. went on to determine the optimal control equations based on the power required for radio frequency transmissions. The problem was developed as a minimax, or game theory, problem meaning that the rover was trying to maximize the transmission power requirement while the relay was trying to minimize the transmission power. By applying Pontryagin's Maximum Principle, a solution set of equations to the problem was obtained. The authors continue from that point to develop a suboptimal solution that is used in solving the solution set of equations and is useful for algorithm development. Most importantly the authors identify that, "the optimal strategy of the Relay is to head toward the midpoint of line segment BO [4, p. 162]." As will be seen in the methodology chapter of this thesis, moving the relay UAV toward the midpoint between the rover UAV and the base, or ground control station, is the control strategy utilized to navigate the relay.

Choi, Pachter, and Jacques continued research with the same model that Pachter et al. defined. They were able to use differential game analysis with optimal control analysis to arrive at a closed form solution. Choi et al. concluded that even in the worst case scenario, as long as the speed of the rover UAV is not more than twice the speed of the relay UAV, all optimal solutions will converge to the relay UAV positioning itself halfway along the vector from the rover to the ground station. The combination of Choi et al.'s research and Pachter et al.'s research provided basis needed to develop the algorithm to navigate the relay UAV [7].

Following Choi et al.’s research, Seibert, Stryker, Ward, and Wellbaum completed the first bench testing of the relay rover communication configuration. In their research effort, the team developed a candidate system architecture for implementation of the relay rover system and the corresponding adaption for integration with other United States Air Force systems (Figure 3). The architecture developed by Seibert et al. is utilized in this thesis, but with modifications. The modifications to the architecture are defined in the methodology section but stem from the limited success that Seibert et al. had in field testing their rover-relay system [6].

Figure 3. OV-1 of Seibert et al. Rover Relay System Concept [6, p. 24]

Seibert et al. built on the cooperative control research of Hansen and Choi with the intent of field testing the rover relay configuration; however, due to the limits of the hardware and proprietary information of the Procerus Technologies Kestrel Autopilot™ system their research team was unable to complete all objectives to fully implement the rover relay concept. The limitations identified were influential for the current research effort because they motivated the change from the Procerus Technologies Kestrel Autopilot™ to the open source Arduino™-based autopilot.

Boire followed the work of Seibert et al. by developing an algorithm to control the relay UAV within the unmodified system architecture that Seibert et al. developed. Boire examined the initial research that Hansen had developed, modified the planar mathematical model, and arrived at the same results concluded by Hansen. Boire found that “an analysis of the instantaneous cost reveals that the midpoint between the ground station source and the rover is the optimal placement of a relay UAV” [5, p. 11]. From this conclusion Boire developed an algorithm that interfaced with Procerus’ Virtual Cockpit™. The basic algorithm function calculated the instantaneous midpoint between the ground station and the rover, and then passed the midpoint global positioning system (GPS) coordinates of that point back to Procerus Virtual Cockpit™ [5]. The algorithm’s functional view, as envisioned in Boire’s architecture, is shown in Figure 4.

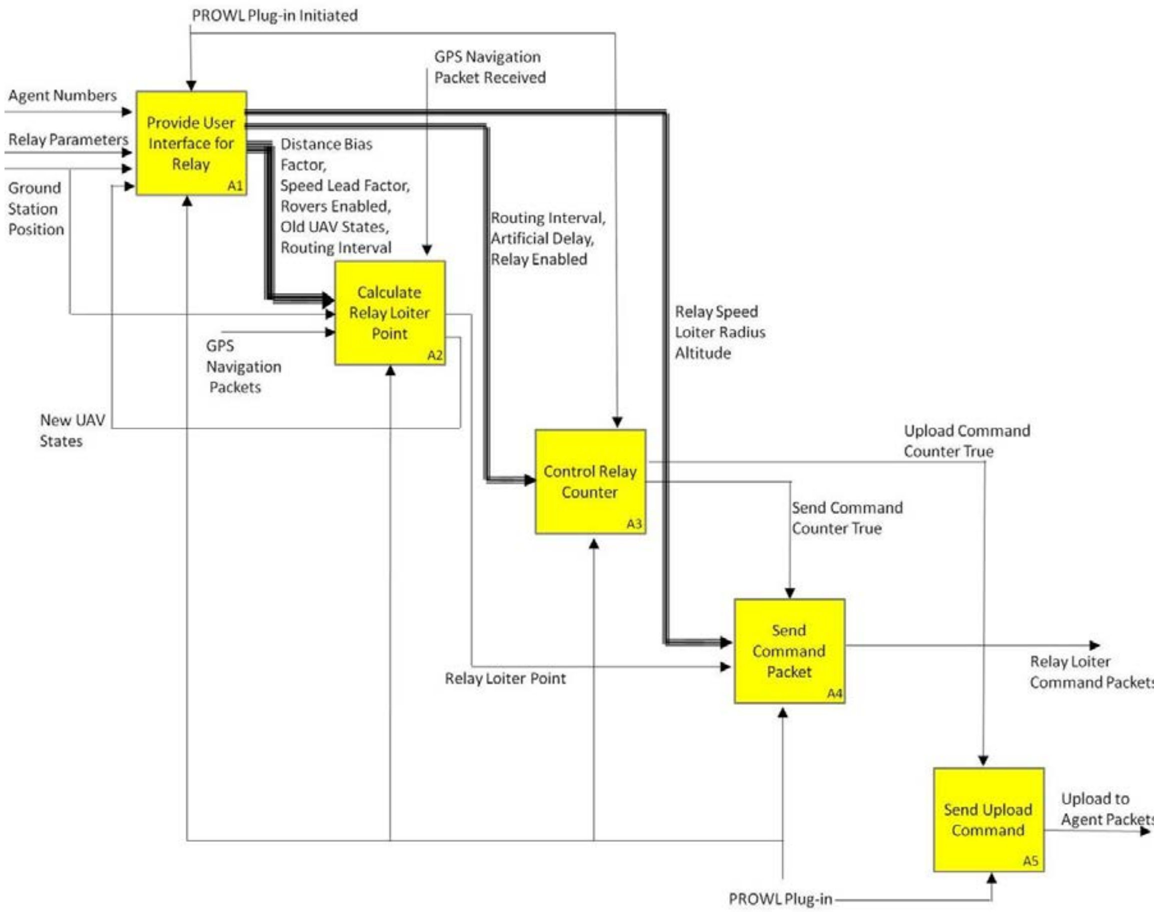

Figure 4. Functional View of Boire's Relay Algorithm [5, p. 22]

Boire ran simulations with the algorithm using Aviones™ and Procerus Virtual Cockpit™. The tests were constructed from combinations of four airspeeds, six loiter point radii, and six routing intervals. Loiter radii are relative to a GPS coordinate; once an aircraft is inside a loiter radius it is considered to have reached the associated navigation point. Loiter radii were created to account for disturbances to the flight path. Loiter radii were varied to examine their effect on optimal flight path navigation. Routing communication intervals were studied to examine the optimal interval to communicate with aircraft. Additional simulations were run to examine time delays, lead compensation (making the algorithm more predictive of where the rover is going), and overall system verification [5].

Simulations indicated that having a relay aircraft that is able to maintain flight at low air speeds and tight turning radii produces more optimal results due to the coupling between loiter radius and relay aircraft speed. As speed is increased, loiter radius must increase and navigation error is induced in the system. A statistical analysis of simulation results indicated that optimal communication intervals should be kept between five and seven seconds. Control input to make the system more anticipatory, known as lead, was examined. Lead compensation analysis indicated that low levels of lead did yield better results; however, lead induced the largest error into the system of all variables examined. The lead compensation projected the future location of the rover UAV by multiplying the instantaneous velocity vector by the time interval between waypoint autonomous generation and by a scaling factor. The lead compensation could be increased or decreased by adjusting the scaling factor. Error was induced in this process because the true flight path was seldom linear. Overall Boire's simulations indicated a potential range increase of 55% over the rover aircraft's original operational range. For a more detailed review of Boire's research please refer to the original document [5].

2.4 Conclusions

There is documented evidence of worldwide demand for UAS technology. The United States Department of Defense and United States Air Force have expressed interest in expanding beyond line-of-sight operations of UAS. AFIT has been conducting research to extend the operational range of SUAS using rover-relay cooperative control since 2008. A mathematical model and initial solutions have been proposed that indicate the relay aircraft should fly to the midpoint between the rover UAV and the ground station to provide maximum operational range of the rover UAV. In addition to mathematical theory, the requirements analysis and system architecture were developed for a candidate rover relay cooperative control configuration. An algorithm was developed, simulated, analyzed and tuned to navigate the relay UAV autonomously for rover relay cooperative control. This area of research is not unique to AFIT, several other researchers have examined similar concepts; however, the focus of this research is scoped to validating AFIT's theoretical advances in SUAS cooperative control [11] [12]. The next chapter will detail the methodology used to build on previous research to develop a SUAS capable of flight testing to validate AFIT's theoretical advances in SUAS cooperative control.

III. Methodology

3.1 Introduction

This thesis is an engineering project targeted at scientific research objectives. As such, a systems engineering approach was selected for guiding principles instead of a traditional scientific method. The “Vee” process model as seen in Figure 5 and described by Forsberg, Mooz and Cotterman, was selected as the systems engineering methodology [8]. Corresponding to the “Vee” process model, the methodology chapter is divided into two major sections. The first section is the decomposition and definition sequence. The second section is the integration and verification sequence. The truncated time table for development, approximately nine months, motivated many design choices. GOTS and COTS components were utilized to shorten the allocation of system functions to subsystems and the detailed design of components phase of the process. The use of GOTS and COTS also allowed the build phase that is usually at the bottom of the “Vee” process to be skipped because the components were already produced. Jumping over the build phase allowed a faster transition to the integration and verification sequence. Testing was integral to the research effort as it spanned the two sequences. Testing identified capability gaps and triggered iterative cycles inside the larger “Vee” process.

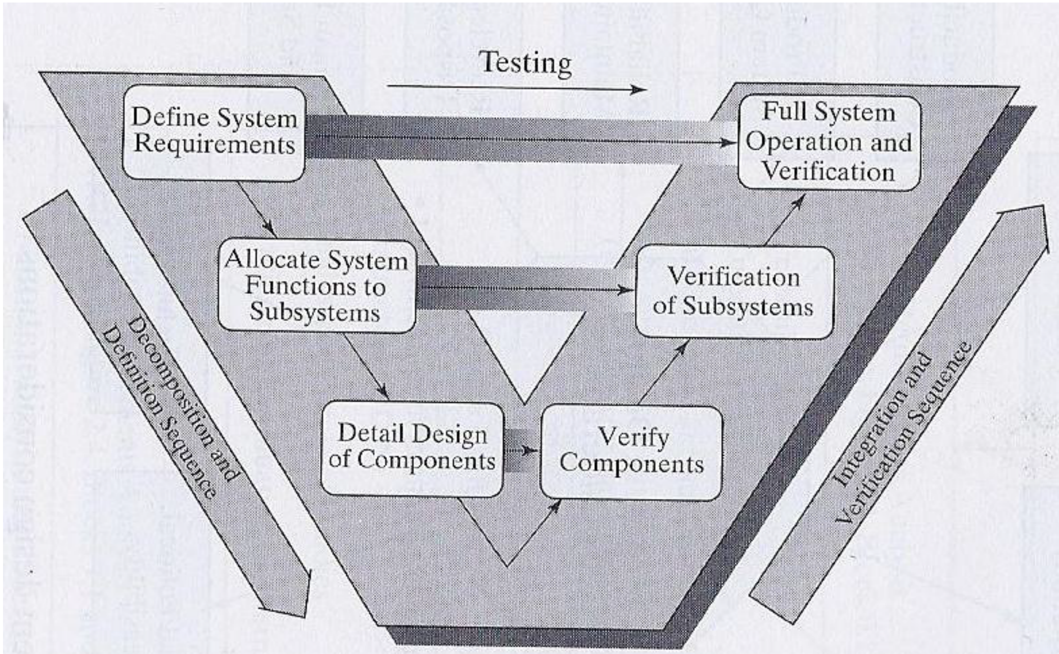

Figure 5. "Vee" Process Model [11, p. 37]

3.2 Decomposition and Definition Sequence

The decomposition and definition sequence is composed of three phases: define system requirements, allocate system functions to subsystems and detail design of components. The sequence started with the original system concept and concluded with the modification and integration activities. The integration and verification sequence follows the decomposition and definition sequence.

The original system requirements were captured in the Unmanned Aircraft Systems Flight Plan. The flight plan identified the need to advance cooperative interaction of SUAS to extend effective line-of-sight operational range [2]. As detailed in the literature review chapter, Seibert et al. examined potential system solutions to meet the primary requirement identified in the flight plan and followed up by developing derived requirements [6]. From a review of Seibert et al., it was determined that an additional requirement is that the system must be based on non-proprietary hardware and software. This new requirement was implemented to avoid the limitations, experienced by past research teams, of proprietary hardware and software from the Procerus™ Kestrel Autopilot™ and Virtual Cockpit™ systems. The switch away from proprietary systems reset the research design from that of previous AFIT SUAS research efforts but still left an initial framework in place. Part of that initial framework specified that a rover relay cooperative control configuration be utilized.

The initial conditions for the design process were: a time table of approximately nine months, a budget that was limited on the order of several hundred dollars of equipment per aircraft (excluding GOTS components), the solution of extending operational range of SUAS using a relay rover configuration, and the requirement to have the relay UAV operate in a transparent manner to the rover operator. Additionally, the airframes that were available as GOTS and COTS options were the OWL and Sig-Rascal 110. The OWL placed size and weight restrictions on the system design. The Sig-Rascal had more than sufficient space and weight available for accomodating the additional hardware. The OWL uses lithium-polymer batteries with an electric motor for propulsion and has a weight of 4.2 pounds, wingspan of 51 inches, and length of 43 inches. A picture of the OWL can be seen in Figure 6. The Sig-Rascal 110 uses a two-stroke engine for propulsion and has a wing span of 110 inches, a length of 52 inches and weight of approximately 14 pounds. The Sig-Rascal is shown in Figure 7. With the project bounded by these requirements, the next task was to allocate functions to subsystems and components.

Figure 6. Overhead Watch and Loiter (OWL) UAV

Figure 7. Sig-Rascal 110 UAV

Allocating functions was expedited by the use of COTS subsystems and components. The time schedule did not allow for development of new hardware components. Additionally, a well established commercial base for micro air vehicles and remote controlled air vehicles provided readily available hardware for the majority of the system functionally needed. The detailed design requirements narrowed the commercially available options to a well-defined set of hardware components. Functional redundancy was kept to a minimum due to the weight restrictions of the available airframes. The act of selecting specific sensors, autopilots, communication components and software determined the allocation of functions. Selection of specific components from commercial options was based on expert opinion of past AFIT SUAS research graduates and the technical support contractor.

Basic commercial components selected for all test vehicles consisted of Ardupilot Mega 2.0 autopilot, MediaTek MT3329 GPS V2.0, airspeed sensor MPXV7002, XBee Pro 900 modem, Castle Phoenix Icelite 50 speed controller, 600mW 5.8GHz A/V Transmitter, FrSky D8R-II 2.4 GHz Telemetry Receiver (ACCST System), FrSky sensor hub and FrSky Lipo Voltage Sensor. Two airframes were utilized: the Overhead Watch and Loiter (OWL) and the Sig-Rascal 110. The OWL is a modified RQ-11 Raven. The original motor and servos were retained in addition to the basic structure and control surfaces of the airframe. The Sig-Rascal 110 was powered by a CCRCPRO GP26R 26.0cc two-stroke engine with a Walbro carburetor and utilized HiTec HS-6635HB digital servos. Once major components were selected and acquired, the next step in the decomposition and definition sequence was initiated.

The detailed design phase consisted of designing modification of GOTS and COTS hardware and open source software to enable integration and functionality of the system. Two significant modifications were to fit the COTS components into the airframe and programming the autonomous loiter point generation capability into the ground control software. Technician support was utilized to design airframe modifications; however, oversight was maintained as a systems engineering function. For programming modifications to the ground control software a programmer was tasked. The basic design requirements of the algorithm were well defined in Boire 2011; however, the algorithm needed modification for integration into the new ground control software [5].

QGroundControl was selected as the ground control software that the algorithm was implemented on. According to the developers, "QGroundControl is an object-oriented C++/Qt application…(that) adheres to the model-view-controller and ISO/OSI layer design patterns" [12]. The developers of QGroundControl specifically developed the software with a modular design to enable extension at each layer of the architecture (Figure 8). The main layers of the architecture are the user interface layer, the Micro Air Vehicle (MAV) abstraction layer, and the Micro Air Vehicle Link (MAVLink) protocol layer.

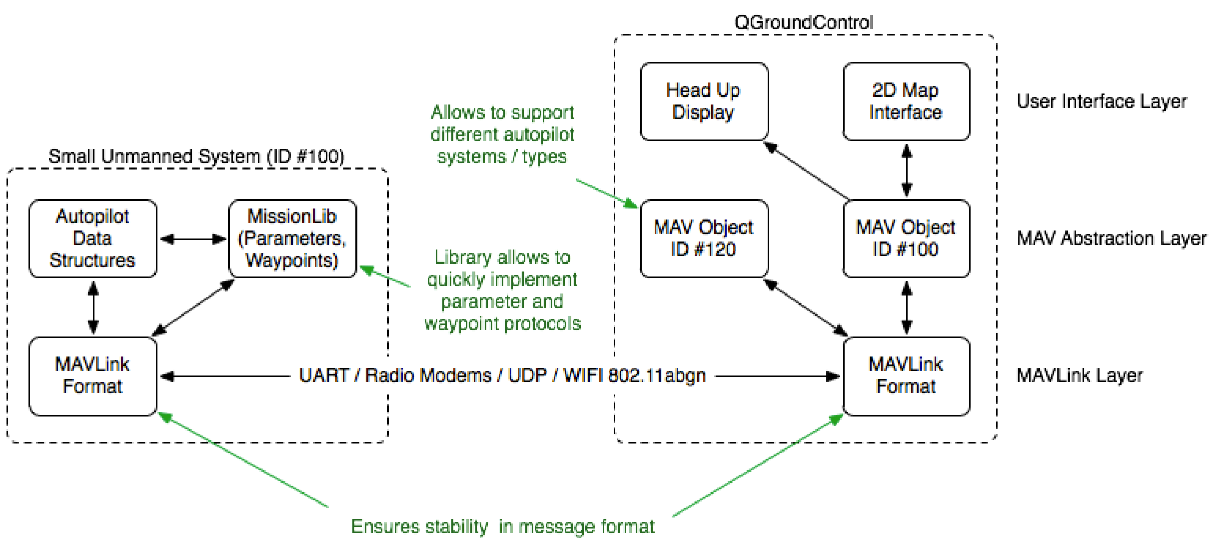

Figure 8. Architecture of QGroundControl [12]

In addition to QGroundControl being developed with future modification in mind, QGroundControl had many native features required for our design. QGroundControl already had the ability to simultaneously read telemetry data from multiple UAVs as long as the UAVs were operating on the same version of MAVLink protocol. The established ability of QGroundControl to handle multiple UAVs simultaneously, in addition to the more common features of telemetry logging, a heads up display, a mission planner, the ability to adjust gains during flight, and the ability to display vital in flight data, kept the ground control software development to a minimum. Hardware integration designs were concurrently developed with the ground control software modifications.

Hardware integration of the various COTS components was the most time intensive work element of the definition and decomposition sequence. First an initial understanding of the basic functional requirements of the Ardupilot Mega 2.0 autopilot (APM) had to be developed. The open source development of the APM meant that there was not a technical support center we could contact for training; instead a Google® hosted wiki and discussion posts from other APM users had to be perused [13]. Just to interface the APM with the ground control software required that the radio control transmitter be powered on and bound to a receiver that was connected to the APM. The APM requires a clean supply of 5.0 +- 0.5 volts. The technical support contractor designed the power supply leaving the integration of components with the APM to be developed. Note that the original design of the power supply did not include power for any video transmitters. This had to be corrected in the next iteration of the power supply design. The original power supply design for the OWL and Sig-Rascal are shown in Figures 9 and 10 respectively.

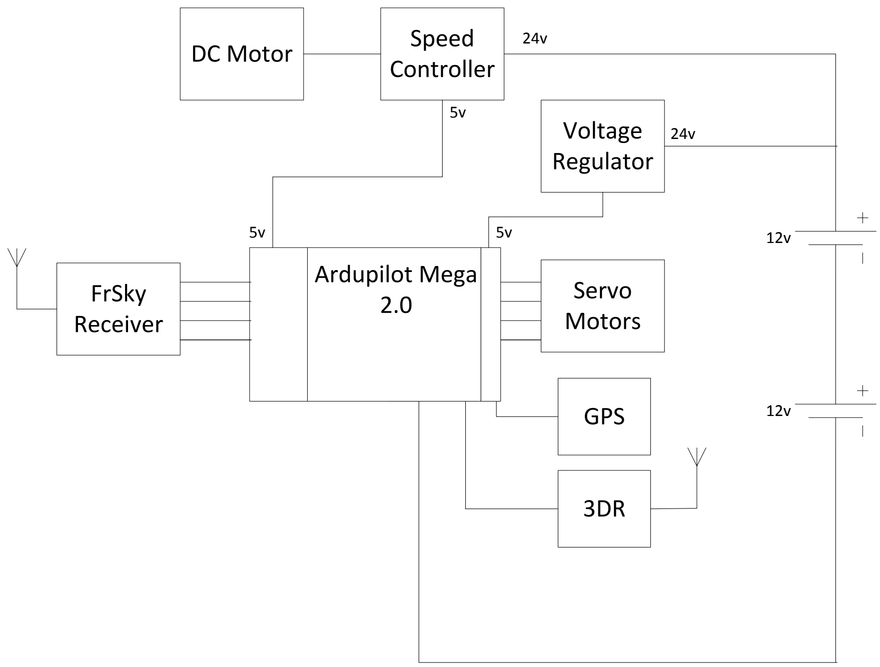

Figure 9. Design Schematic of OWL [16]

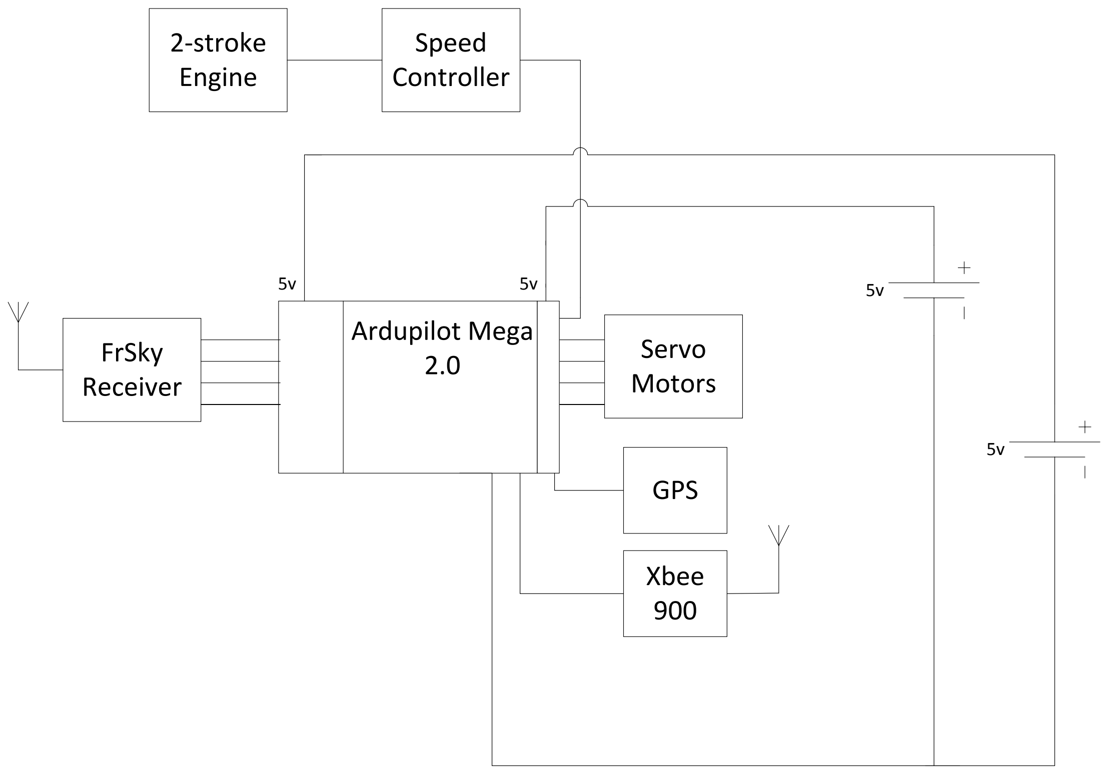

Figure 10. Design Schematic of Sig-Rascal [16]

The APM is developed to be adaptable to multiple airframes. As can be seen in Figure 11, each bus has an intended use; however, the component connected to any given pin set is specified in the firmware. It is important to note that on all busses the outside pin is ground, the middle pin is five volts and the inside pin is data. Figures 12 and 13 show which component connected to each utilized pin set. The input bus pin set layout matches the output bus pin set layout with one exception. The input bus has an additional pin set to allow the Radio Control (RC) operator to set the mode the aircraft is operating in. For this design, channel eight was used to control the autopilot mode.

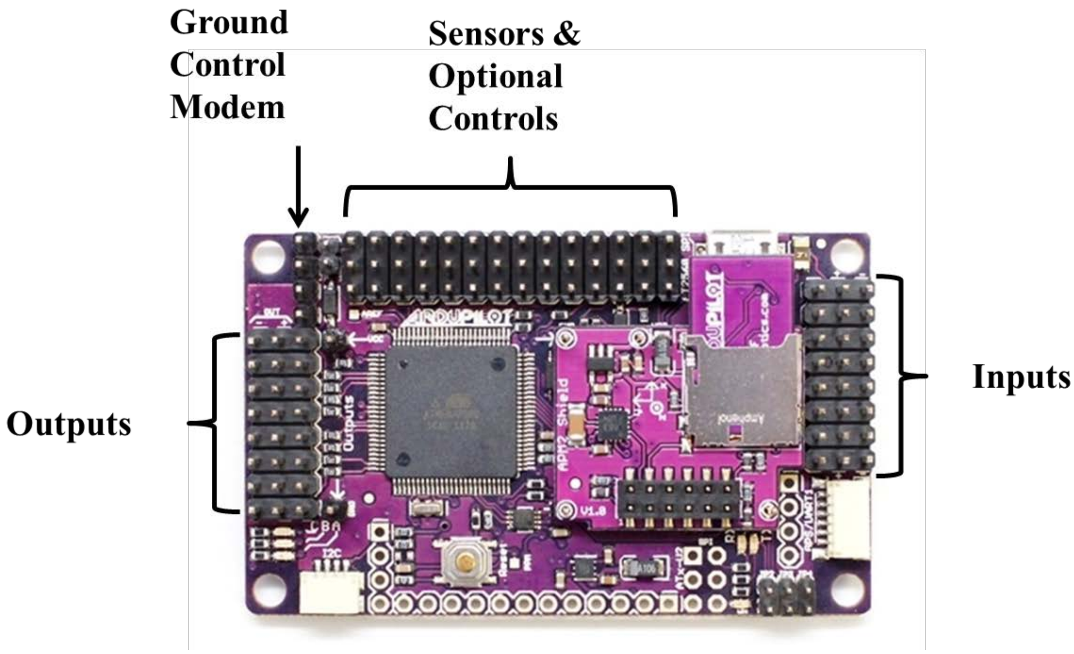

Figure 11. APM Board with Busses Labeled

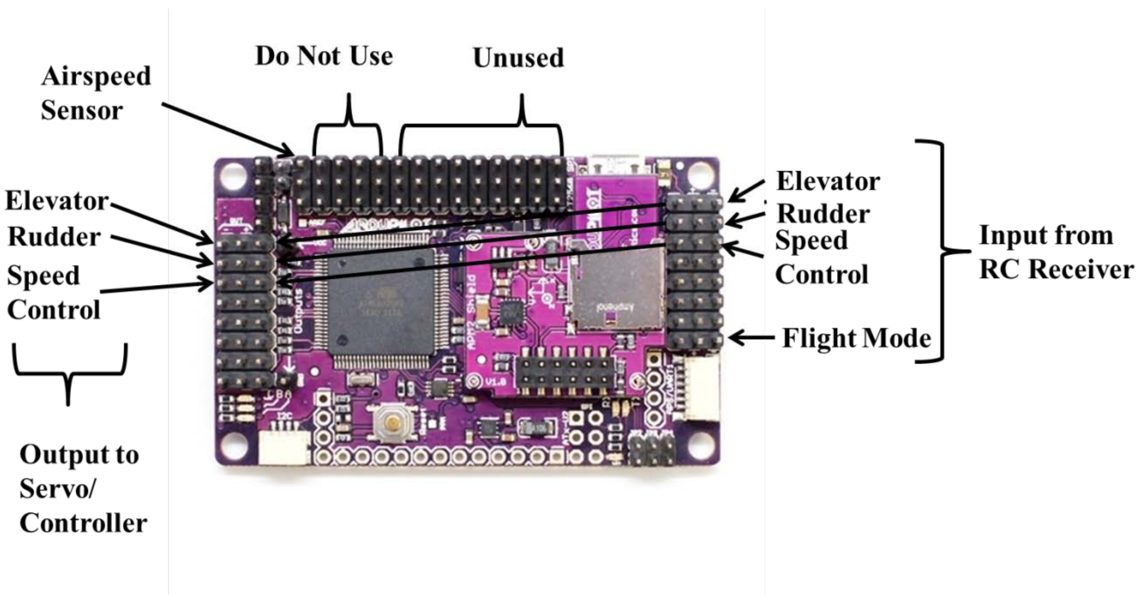

Figure 12. OWL Pin Set Layout

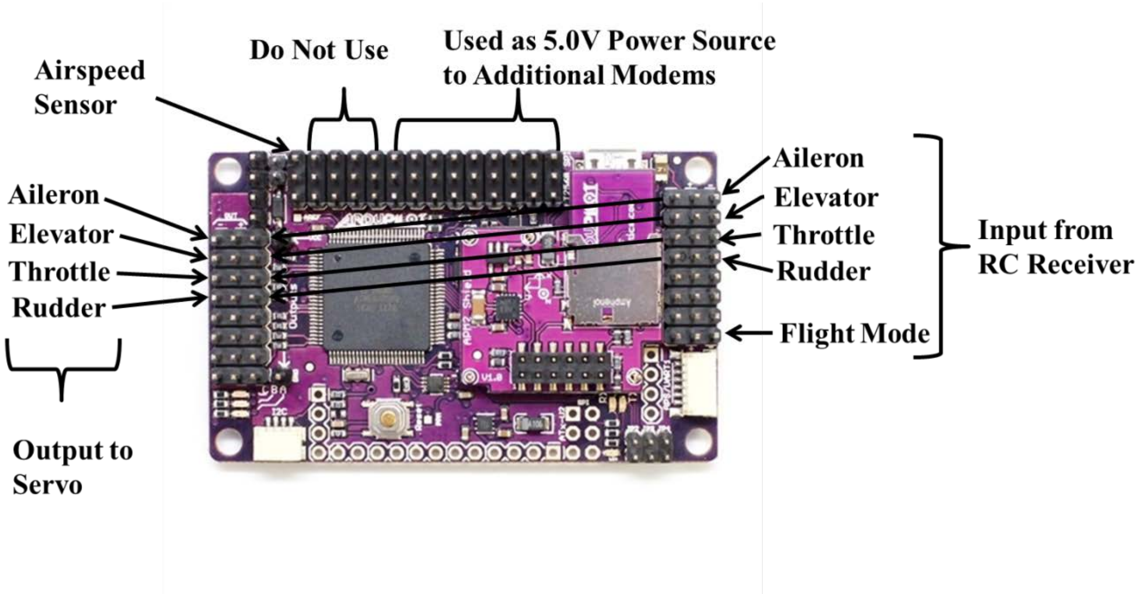

Figure 13. Sig-Rascal Pin Set Layout

At this point in the definition and decomposition sequence of the design cycle enough detailed design decisions had been made that initial integration and verification actions were commenced. This was not recognition that all decomposition and definition activities had been completed but recognition that enough progress had been made to test basic functionality of integrated components. The goal was to integrate enough components to conduct initial flight tests. Flight test procedures were developed, each step successively isolating one capability before moving on to combined capabilities. See Appendix A in the initial flight testing section for detailed flight testing procedures. These initial tests results helped to keep the decomposition and definition sequence from building on poor or inoperable design choices.

Initial test results revealed that while many of the designed capabilities were functional, not all components were integrated successfully. The original 915 MHz 3DR radio utilized as the modem for ground control software to communicate with the UAV was incompatible with QGroundControl. This was realized early enough in the project timeline that a different modem (XBee 900 Pro) could be purchased and integrated into the design. Using the original 3DR modem and an alternative ground control software called Mission Planner, flight testing was conducted [13]. These preliminary flight tests revealed that enough operability was developed to tune the autopilot, write mission plans to the autopilot, and fly the OWL platform in autopilot mode. The procedure for tuning the gains for the autopilot are detailed in Appendix B. Additionally, testing indicated that the integrated current and voltage monitoring capabilities of the APM 2.0 autopilot were not reliable enough for purposes of this research project. While it was a goal of early flight testing to fly multiple UAVs simultaneously, these test objectives were not met due to the incompatibility between the only modems on hand at the time of testing and the ground control software. This made ground control software integration a high risk element in the project.

Given the initial test results, an assessment of the major risks to the project was conducted. Already it was clear that modem compatibility with the ground control software could be an issue. The lack of early multiple vehicle testing meant that we did not have data to indicate if the design choices made regarding QGroundControl were functionally able to be integrated with the other components in the system. These factors made ground control software stand out as a prominent issue in the risk assessment. The successful flights in both manual and auto modes of the aircraft reduced many other risk factors such as component integration, component functionality, and weight distribution.

The risk caused by the inability to obtain satisfactory voltage data from the batteries during flight was mitigated by integrating the FrSky voltage sensor into the design.

The top five remaining risk elements are shown in Figure 14. Identifying the top five remaining risks enabled enacting risk mitigation efforts. Gound testing of the XBee Pro modem and QGroundControl software was scheduled and conducted to identify integration and functionality issues earlier in the design process. Additionally multi-UAV bench testing was scheduled to mitigate the risk of flight testing revealing problems too late in the design process. Having time to address the integration and functionality issues reduced the risk. The risk of test range scheduling was assumed without mitigation efforts because utilizing an alternate test range was not within the budget resources available. The risk of QGroundControl not being well documented was also assumed because QGroundControl was the best documented ground control software for the APM 2.0. Knowing the risks the project was susceptible to, the decision was made to continue with the decomposition and definition sequence.

Research Risks

Likelihood		Insignificant	Minor	Moderate	Major	Catastrophic
	0.9			5		
	0.7			3	2 4	
	0.5		1			
	0.3					
	0.1					
		Consequence				

#	Risks:
1	If XBee Pro modem is incompatible with Ardupilot, then communication range and network configurability will be limited and time will be lost.
2	If QGroundControl software is not functional with our UAV hardware, then multi-UAV single-GC configuration will not be possible, causing a new requirement for GC-GC networking.
3	If Atterbury range isn't available for flight test scheduling, our flight tests will slip, possibly causing us to lose flight tests.
4	If QGroundControl is not well documented, then modifications to the software will require more time and could be impossible on our schedule.
5	If multi-UAV bench and flight testing is not completed early, then the potential increases for unseen requirements to develop late in the project development.

Key

Low Risk

Moderate Risk

High Risk

Figure 14. Project Risk Chart

Following the preliminary flight testing, the last piece of the design that needed to be defined was to capture the complete picture of requirements needed for integrating the algorithm. To capture the requirements for integrating the algorithm with the ground control software a typical mission profile was examined. Specifically, a mission profile was examined by developing an operational architecture process flow diagram. Figure 15 details the process flow for operations. The diagram was useful for identifying how the autonomous control algorithm must interact with the changing states of the ground control software from initially connecting to the UAV to landing the UAV after a completed mission. By examining the process flow for operations it was determined that the ground control software must be modified to be able to: identify one UAV as the relay aircraft, identify one UAV as the rover aircraft, calculate the midpoint between the ground control station and the rover aircraft on a specified interval, write the midpoint location as a loiter point for the rover to fly toward on the same specific interval, and have the ability to disable the autonomous navigation algorithm for launch and landing situations. With the specific functional requirements defined the next step was to meet with the programmer.

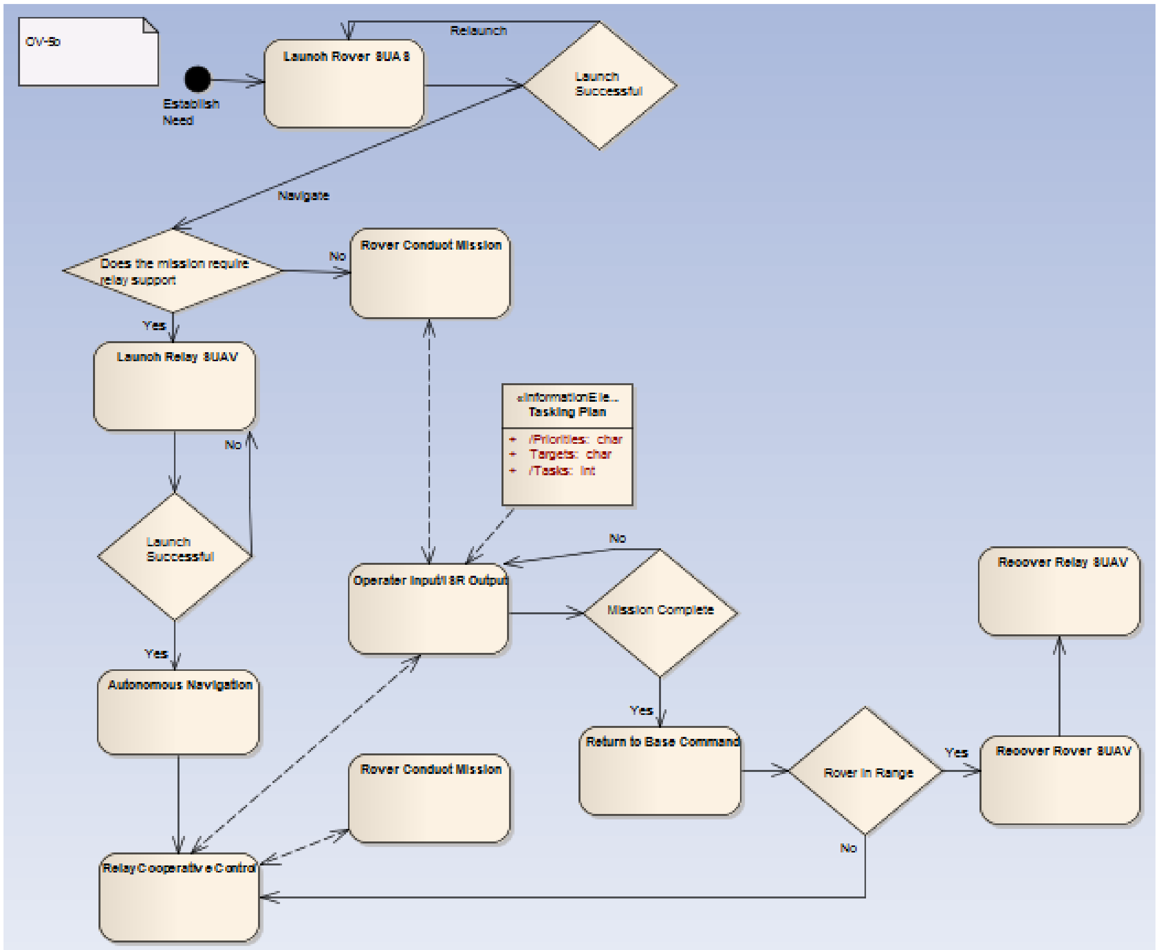

Figure 15. Process Flow for Conducting Mission (OV-5b)

The requirements for the modification of QGroundControl were presented to the programmer. The programmer assessed that while the additional requirements were not impossible, our development schedule and resources available were not adequate to fully develop the requested functionality. One major issue identified was that while QGroundControl has the native ability to simultaneously update telemetry data from multiple UAVs, it can only have one UAV selected for active control at any given instant. This meant that the objective of having one operator flying the rover UAV in extended range just as she/he would in normal operating range, with the relay operating transparently, could not be realized on the same instance of QGroundControl without prohibitively major modifications to QGroundControl. Additionally the autonomous time interval for calculating the midpoint and writing a loiter point was assessed to be too complex for schedule and resource limitations. The autonomous time interval is the duration of time between cycles of generating new waypoints. These limitations forced a re-analysis of the core requirements necessary just to achieve the technology demonstration of rover relay configured extended line-of-sight operations.

Simplified requirements, or test expedient requirements, to demonstrate the technology required the ground control software to be able to simultaneously update two UAV telemetry data streams, calculate the midpoint between the rover and the ground station, and write the midpoint as a loiter point to the relay UAV. The idea was proposed to use a separate ground station for the rover UAV. QGroundControl would read the telemetry of the rover UAV and the relay UAV but would only control the relay UAV. The relay specific version of QGroundControl would be modified to operate only in relay mode. The safety pilot would have to take manual control of the relay UAV for any flight time not pursuing the midpoint. By removing the requirement for automation of the interval for calculating the midpoint, a requirement for an additional operator that would initiate a mouse click event in place of the automation was added. Figures 16 and 17 below show the different architectures for the original requirements and the test expedient requirements for the ground control station.

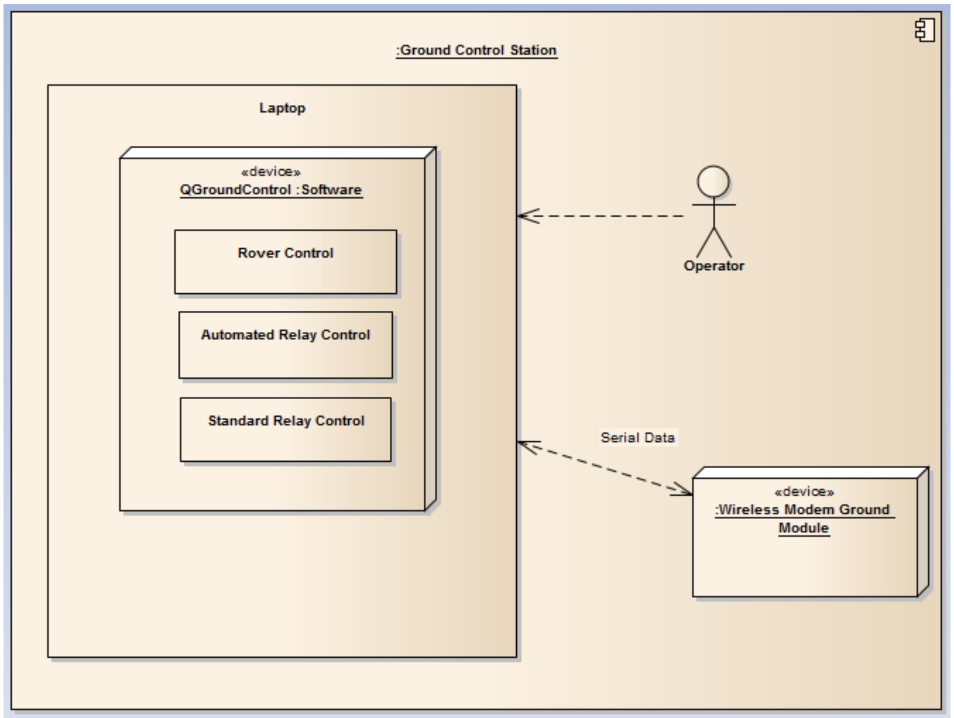

Figure 16. Original Ground Station Architecture

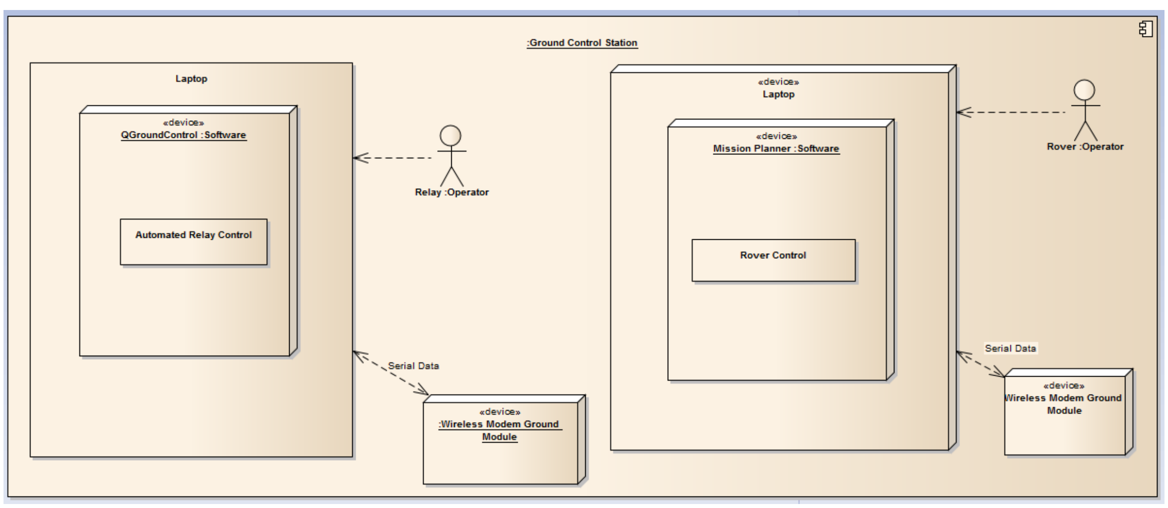

Figure 17. Test Expedient Ground Station Architecture

The original analysis Boire completed for midpoint calculation was preserved in the midpoint calculation algorithm; however, given the conclusions of Boire's simulations, adaptations were made for implementing the algorithm. Lead compensation was not designed into this implementation of the algorithm [5]. For the autonomous navigation to have all necessary data to generate the loiter point for the relay aircraft to navigate toward, the MAVLink protocol required that latitude, longitude, altitude, and loiter radius be set [12]. It was determined that test objectives could be more readily achieved if the gas powered Sig-Rascal 110 airframe was utilized as the relay platform because it would benefit the ongoing research of Songer [16]. Using the Sig-Rascal as a relay would allow more weight and more space for communications equipment utilized for the actual communication relaying. Using Boire's simulation data for communications signal strength optimization, loiter radii were coded to 80 meters given that the cruise speed of the Sig-Rascal as configured for flight testing was 18 meters per second. Due to the test expedient design compromise of not being able to turn on and off the automatic waypoint generation, it was decided that a standard altitude of 100 meters above the altitude of the flight test range would be utilized. This design decision reduced communication signal strength optimality of the algorithm but increased the safety of flight testing. It would have been more optimal to have the relay UAV fly at an altitude half way between the altitude of the ground control station and altitude of the rover UAV; however, reducing the risk of flying the relay UAV into the ground autonomously if the rover lost altitude was deemed more important than the reduced optimality. Finally the process for integrating the algorithm to determine the latitude and longitude needed to be defined.

Implementation of the algorithm was motivated by simplicity of programming due to the time and resource limitations of the design effort. The algorithm was developed internal to QGroundContol such that the first time the operator clicked on the mission planner map it would set a waypoint numbered one. This waypoint would need to be the location on the map where the ground control station was located. Waypoint one would be used in the algorithm to extract the latitude and longitude of the ground control station, commonly referred to as ‘home’ location in the QGroundContol developers terminology [12]. Next the operator would double click in any location on the mission planner map. The algorithm would automatically calculate the desired midpoint between the ground control station, waypoint one, and the latest latitude/longitude telemetry data that QGroundControl had for the location of the rover UAV. To meet the native waypoint protocol structure within QGroundControl, an additional waypoint had to be generated beyond the midpoint loiter point. The additional point would never be navigated toward and could thus be arbitrarily selected. It was decided that the location of this arbitrary waypoint would be the latitude and longitude of the rover UAV utilized for midpoint calculations because when the loiter point and arbitrary waypoint were generated on the map, it was simple to visually reference if the calculations appeared accurate.

3.3 Integration and Verification Sequence

At this point in the project, all major decomposition and definition sequence activities had been completed and the focus of the project became actions of the integration and verification sequence. The integration and verification sequence is composed of verify components, verification of subsystems and full system operation. As noted in the discussion of the decomposition and definition sequence, preliminary integration and testing activities had been conducted. This preliminary testing led to the integration of additional components.

The additional functions with their corresponding components; namely the FrSky voltage sensor, FrSky senor hub, XBee Pro 900 modem, and modified QGroundControl; still needed to be verified but all other functions with their corresponding components had been verified. The additional components were self-contained subsystems so the activities of component verification and subsystem verification were conducted simultaneously. Components were verified in the lab to ensure they met the requirements and performed as anticipated. The voltage sensor and sensor hub were powered on following FrSky's instruction and voltage data was properly displaying on the safety pilot's radio [13]. The XBee Pro 900 modem was tested by establishing communications between two XBee Pro 900 modems on development boards. The modified version of QGroundControl was tested using software in the loop testing (SIL). SIL testing utilized a built in simulation intended to demonstrate the capabilities of QGroundControl. The loiter point, home location and additional waypoint were generated in the mission planner. The functionality of writing the waypoints to the UAV could not be tested during component/ subsystem verification because such testing required integration with the full system.

The next activity of the integration and verification sequence was full system operation and verification. The voltage sensor, sensor hub, and modem were installed in the UAVs. Integration was completed with the installation of the additional components.

Figures 18, 19 and 20 show the internal components of the OWL UAV with the body panels and wings removed from the airframe.

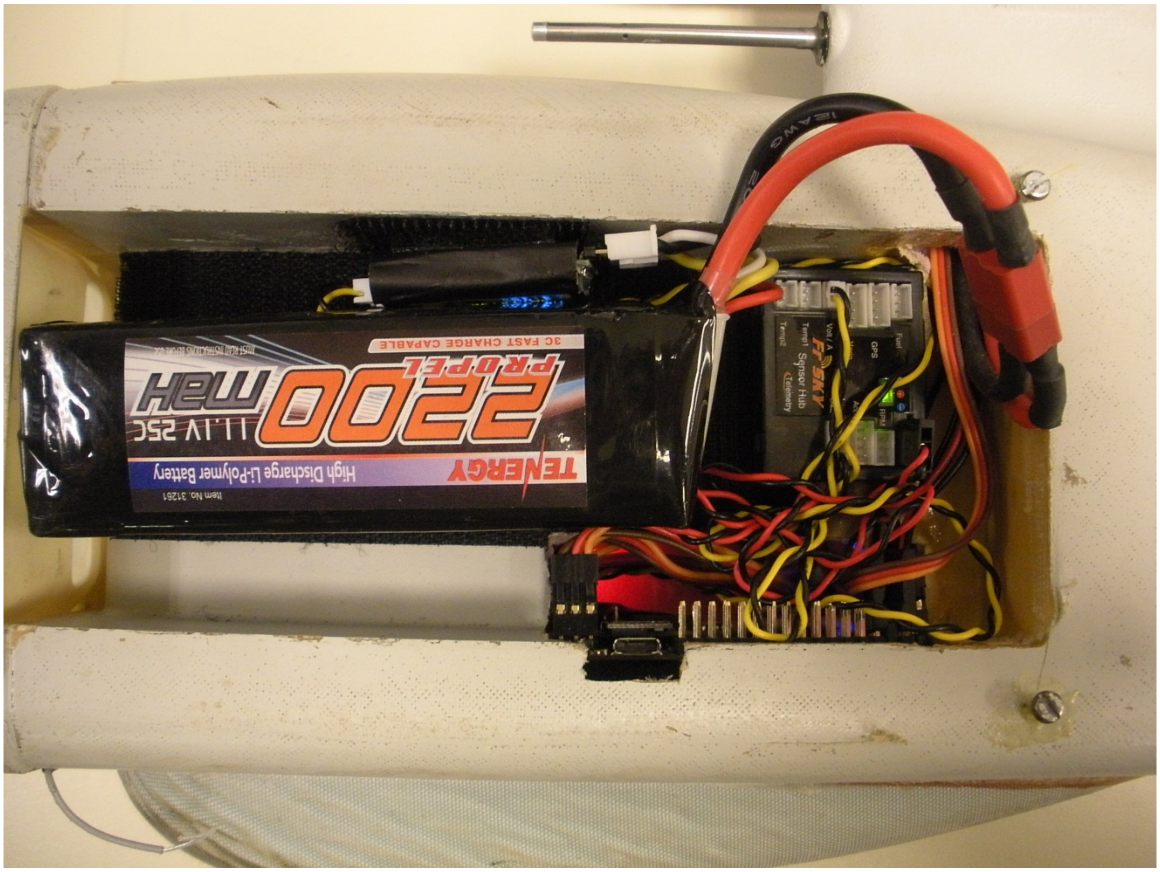

Figure 18. OWL Left Side View

Figure 19. OWL Right Side View

Space for components inside the airframe was a limited resource as can be easily seen in Figures 18, 19 and 20. In Figure 18 the left side battery, voltage sensor, sensor hub, USB socket and sensor and optional control bus are visible. In Figure 19 the right side battery, RC receiver, video transmitter and output bus are visible. In Figure 20 the GPS, electronic speed controller, and combination static and dynamic pitot tube are visible. With the body panels, wings, nose cone and tail attached the fully integrated OWL airframe was ready for system verification.

Figure 20. OWL Top View

The Sig-Rascal 110 was simultaneously assembled with the OWL for full system integration and verification. Figures 21 and 22 show the Sig-Rascal with the wings removed. In Figure 21 one of the two relay modems antenna and a third ground control modem antenna are visible. Three modems had to be integrated in the design because an attempted mesh network modem did not provide the necessary functionality, see Songer for more details [16]. Additionally, in Figure 21 the prop, muffler, clear gas tank panel, battery voltage indicators, and external power switches are visible. Figure 22 shows a top view of the APM 2.0 as integrated with the Sig-Rascal. Additionally the RC receiver and two relay modems are visible in Figure 22. With the wings attached the Sig-Rascal 110 was ready for full system verification.

Figure 21. Sig-Rascal 110 with Wings Removed

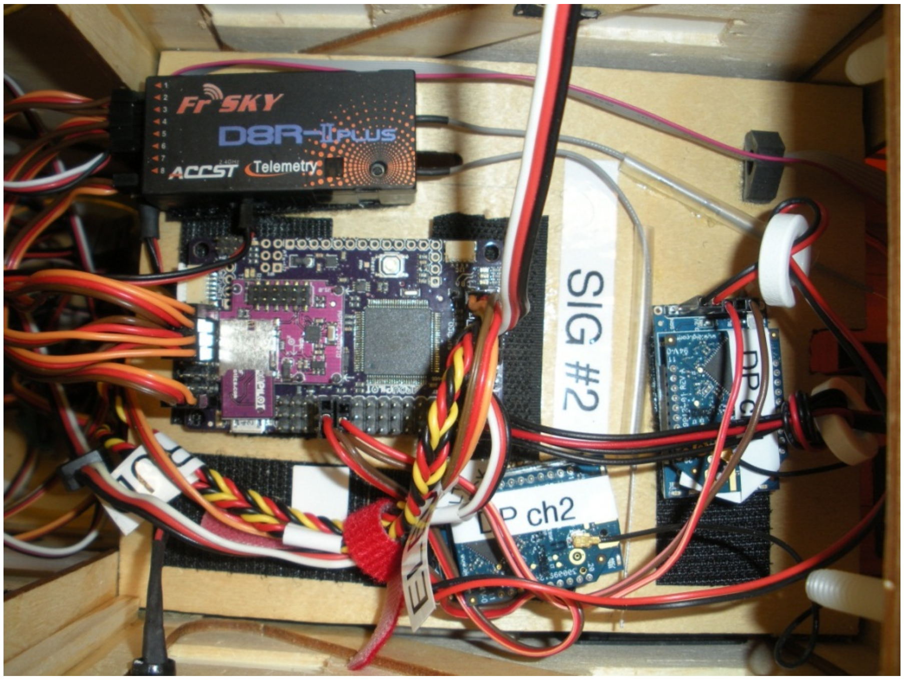

Figure 22. APM 2.0 as Assembled in Sig-Rascal 110

Ground testing was begun for full system operation and verification. Ground testing followed the exact same procedure as flight testing except the prop was removed from the

OWL and the engine for the Sig-Rascal was not started. Instead of flying, the UAVs were driven around on a golf cart for ground testing. The motor would spin and the control surfaces would respond as various commands were given to the autopilot. Flight testing procedures are detailed in Appendix A. A basic description of the tests and objectives is given in Table 1.

Table 1. Basic Test Description with Test Objectives

Test	Objective
Initial communication check	Prior to flight just make sure each UAV is functional
Initial single UAV flights (Using Mission Planner)	Fly each aircraft to verify functionality, adjust trim, & tune gains
Initial single UAV flights (Using QGroundControl)	Make sure unmodified QGC is functional
In flight range check	Determine maximum range of single point to point modems
Multi-UAV Flight	Verify ability to fly multiple UAVs simultaneously
Multi-UAV Flight with relay within direct range	Verify ability to relay signal Verify autonomous navigation
Multi-UAV Flight with relay (BLOS)	Full system verification

What was not understood at the time of ground testing is that the APM 2.0 is supposed to use the airspeed sensor to determine the state of the UAV. If the airspeed is below some threshold the autopilot is supposed to know it is not flying and should not attempt to navigate autonomously. Despite the fact that low airspeed was registered during ground testing, the motor and control surfaces still responded to ground test inputs. The airspeed restriction on the state of the autopilot was not understood at the time of ground testing. The ground testing objective was to verify that the fully integrated system appeared operational. The operational status was difficult to discern because the aircraft responded to input but since flight did not occur it was not clear if the aircraft response was what it should be. At a minimum both airframes were responsive to inputs during ground testing.

Flight testing required a substantial support structure. Flight testing was conducted at Camp Atterbury in Indiana. The technical support contractor provided power generators, a ground control trailer, field repair expertise, and the RC safety pilot. Weather restrictions for the OWL airframe limited the operational envelope to exclude precipitation and winds that gusted over 15 miles per hour. The tower at the airfield provided the weather condition information to determine if the weather requirements were met. Figures 23 and 24 show the flight testing conditions.

Figure 23. Flight Testing Ground Control Station

Figure 24. Sig-Rascal 110 During Take Off

3.4 Conclusions

The systems engineering “Vee” process model provided a structured approach to the engineering project. Iterations of decomposition and definition coupled through testing with iterative integration and verification kept the project from building on incompatible design decisions. A continued effort to scope the project within schedule and resource constraints required careful management of requirements. Careful management of requirements kept the focus exclusively on what constituted capability minimums to demonstrate the technology of rover relay cooperative control to extend SUAS line-of-sight operations. Test results, discussed in the next chapter, indicated the degree of success this project was able to achieve.

IV. Results

4.1 Introduction

This chapter details the capabilities demonstrated as a result of the engineering project. The full scope of the objective originally defined for the engineering project was not fully attained; however, partial functionality of the SUAS was demonstrated in flight testing. The goal of the research was to develop, integrate and implement system hardware and software necessary to validate AFIT's theoretical advances in SUAS cooperative control. The attempted end state objective of the research effort was to flight test an autonomous control algorithm on a relay UAV that was actively relaying data to a rover UAV in an extended line-of-sight operating range. A successful transition was achieved from previous proprietary test systems to an open source test system based on the APM 2.0. Flight testing demonstrated the SUAS's ability to generate the correct navigation data autonomously; however, the navigation data was not successfully activated as current waypoints on the relay UAV's autopilot. Software in the loop testing was utilized to verify a solution to make the navigation data be the current waypoint but flight testing was not conducted to verify the simulation results.

4.2 Test Results

Preliminary flight testing was able to demonstrate that integration was successful enough to conduct manual and autopilot flight missions. The preliminary flight testing also resulted in changing the modems used for ground control station communications and the addition of voltage sensors to the UAV design. With those changes integrated into the design the next round of testing completed was the full system verification.

Full system verification yielded partially successful results, 94%, as can be seen in Table 2. The capabilities are across the top of the Table and the components that enable those capabilities are down the side of the table. If an ‘X’ is in the box at the intersection of the components and capabilities then the component is needed to enable the capability and was verified to be operational in flight testing. If an ‘O’ is in the box at the intersection of the components and capabilities then the component is needed to enable the capability and was not operational in flight testing. If the box at the intersection of the components and capabilities is blank then the component is not needed to enable the capability. Most noticeably missing from the table is the communications relay capability. The communications relay capability was the focus of Songer’s research. For a more detailed analysis of communications relay results please refer to Songer’s thesis [16]. The 94% success rate was determined by dividing the number of verified capabilities by the total number of needed including the relay capabilities.

Note that while the capabilities of flying in-flight programmed waypoints and autonomous waypoints were not demonstrated, some of the lower level requirements culminating in those capabilities were successfully demonstrated. Flight testing confirmed that the correct calculations were made and the correct waypoint data was able to be sent to the relay UAV.

Table 2. System Capabilities for OWL Platform

	Navigation				I.S.R.	Flight		System Status
	Fly Pre-programmed Waypoints	Fly In-Flight Programmed Waypoints	RC Pilot	Autonomous Navigation	Video	Normal Operations	Emergency Operations	In Flight Battery Voltage Data
Ardupilot Mega	X	X	X			X	X	
GPS	X	X				X	X	
Data hub								X
Speed controller	X	X	X			X	X	
Servos	X	X	X			X	X	
Airspeed sensor	X	X				X		
Batteries	X	X	X			X	X	X
Camera					X			
Video Transmitter					X			
Radio Control Receiver			X			X	X	X
Radio Controller			X			X	X	X
914 MHz Modem	X	X		X		X		
Ground Control Computer	X	X		X		X		
Ground Control Software	X	O		O		X		
Autonomous Way Point Algorithm		X		X				

Legend	
Symbol	Meaning
X	Linked & operational
O	Linked & not operational

The limitation in achieving the two capabilities, the ‘O’s in Table 2, was identified to be the inability to activate the new waypoints of each interval to update the midpoint. Once the aircraft was launched and turned over to autopilot control, no flight test data indicated the ability change the active navigation points on the autopilot. Following flight testing, software in the loop testing was utilized to verify a solution to change the active waypoints in flight. Additional flight testing to verify the results of software in the loop testing was able to be completed.

In addition to having the capability to fly in autopilot mode, each airframe had to have a set of gains adjusted to have the autopilot mode function properly. The gain tuning procedures are documented in Appendix B. Gain tuning was necessary to enable the primary flight testing objectives but was not a direct research objective so a technician’s tuning procedure was applied instead of a more in-depth analysis. Gains for autopilot flight of both the OWL platform and the Sig-Rascal platform were obtained and can be seen in Figure 25 and Figure 26 below.

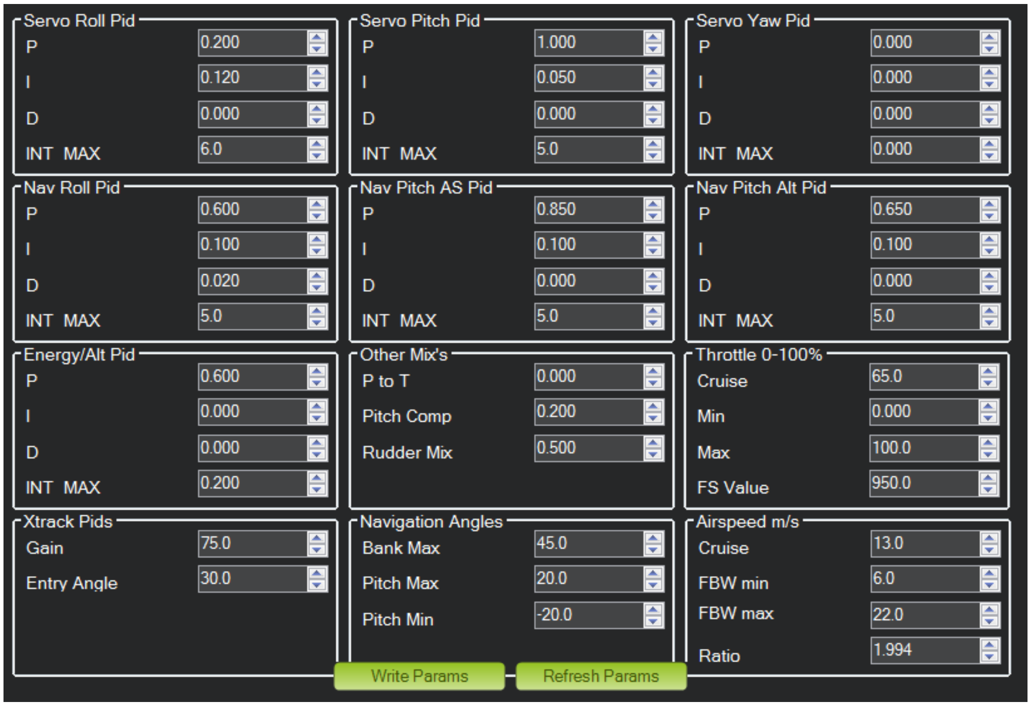

Figure 25. Gain Parameters for OWL Platform

Figure 26. Gain Parameters for Sig-Rascal Platform

Full system verification through flight testing also revealed that a redesign of the power supply on both the OWL and Sig-Rascal airframes was required. The original design of power supply resulted in the APM 2.0 board power cycling during flight because two switching voltage regulators were integrated in parallel causing power anomalies. The redesign utilized one switching voltage regulator to power the entire APM 2.0 using a jumper to connect the power from the output rail to the input rail. The redesigned power supply for the OWL and Sig-Rascal is shown in Figures 27 and 28.

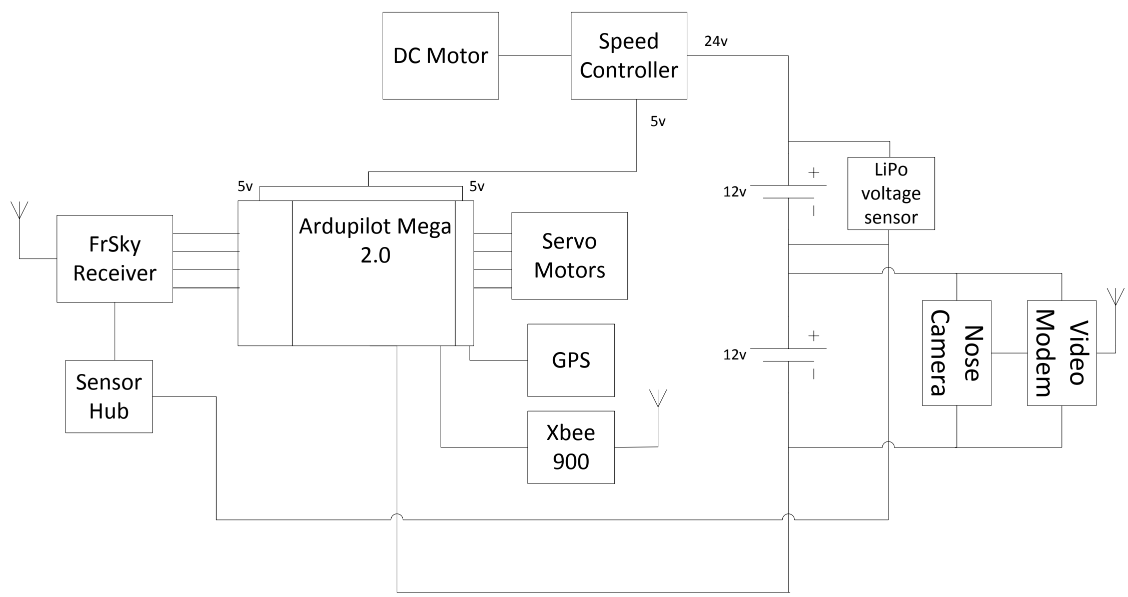

Figure 27. Redesigned OWL Schematic [16]

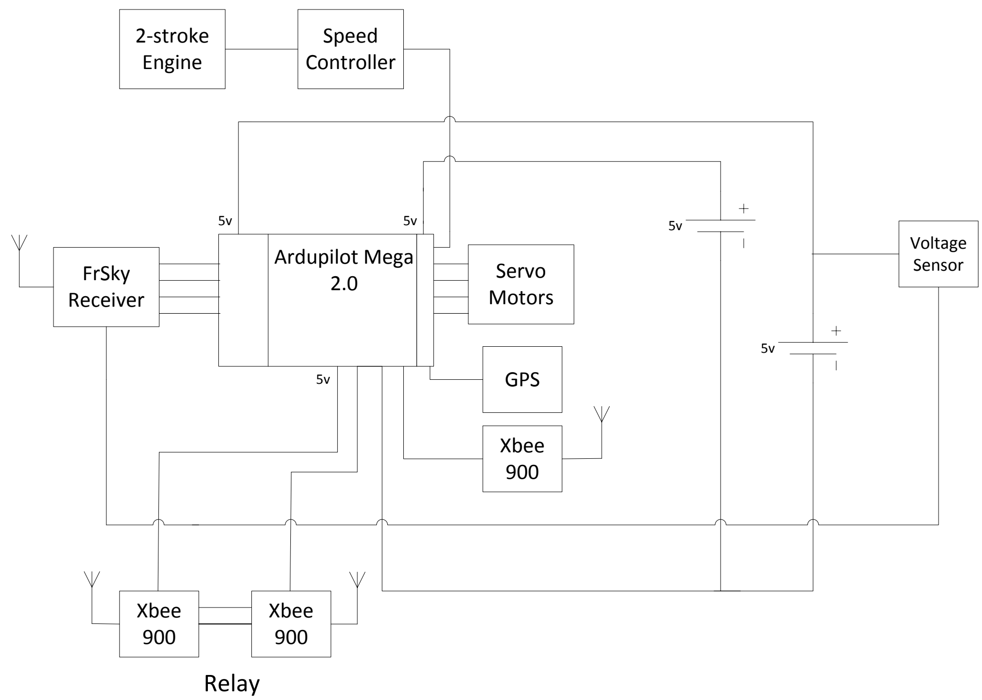

Figure 28. Redesigned Sig-Rascal Schematic [16]

Following flight testing, programming was completed to add the ability of activating the autonomously generated waypoints to be the current waypoints on the autopilot. Software in the loop testing was used to verify that the solution developed actually worked. Since follow-on flight testing to verify the solution was not able to be completed in the timeline of the project, the ability to demonstrate autonomous waypoint navigation is not claimed as a success. While flight testing could reveal additional design modification necessary to demonstrate autonomous control algorithm, the software in the loop testing indicates that the only step needed to be completed is to conduct another round of flight testing. Additionally Songer was able to implement design modifications that demonstrated the relay communications are operational in ground testing [16].

4.3 Summary

While significant progress was made toward establishing an open source test platform, the attempted end state objective of the research effort was to flight test an autonomous control algorithm on a relay UAV that was actively relaying data to a rover UAV in an extended line-of-sight operating range. Neither the autonomous control algorithm nor the actively relay data objectives were successfully flight tested. Work following flight testing has indicated that both the autonomous control algorithm and the relay objectives are ready for another round of flight testing. The resources and operational weather conditions to complete the flight testing were not available at the time of the completion of this thesis. Flight testing did demonstrate important enabling functions toward the objectives. The autonomous control algorithm was able to calculate the correct midpoint loiter point and was able to write the home location, loiter point and additional waypoint to the relay UAV. The UAV was given all the navigation input necessary to fly to the correct location for relaying the signal; however, the data was never activated for navigation in flight testing.

V. Conclusions

5.1 Chapter Overview

This chapter entails two discussions. The first discussion is about the systems engineering process applied to this project. The “Vee” process model was beneficial in guiding the engineering project and many guiding systems engineering principles were successfully applied throughout the project. The second discussion examines the future work building on the technology demonstrated from the flight testing results.

5.2 Conclusions

The “Vee” process model was a useful guide in this design project. The area the “Vee” process model helped the most was in keeping design development focused on the requirements. This was critical for the project because of financial and, more importantly, time resource restrictions. The intermediate testing prescribed by the “Vee” process model was what kept the design on track for integration [8]. It was the structure of the “Vee” process that helped achieve the 94% functionality success because the focus was kept on requirements.

Unfortunately, it was intermediate bench testing that yielded false positive results that QGroundControl was fully operational. The act of writing the waypoints to the UAV did not activate those waypoints for navigation. It is a systems engineering principle that testing be conducted as close as possible to the intended operational environment. The bench testing to verify the ground control software was ready for flight testing was simply not tested in a manner close enough to flight testing conditions. If it had been tested in an environment more realistically representing the flight testing conditions, the inability to activate waypoints would have been detected in time to correct the oversight before flight testing. Instead, the full capability of the autonomous control algorithm was not flight tested because of an oversight about requiring waypoint activation. The structure of the “Vee” process was not at fault for this error. The “Vee” process was a success as applied in this project despite the challenges that existed.

The largest challenge to this research project came from the open source aspect of the project. The open source software really turned out to be an important design trade off. The nature of the open source software allowed access and modification at all levels of the design. In the scope of the project, the advances demonstrated were partially the result of new designs for component integration; however, most of the advances came as a result of modifications made to the open source ground control software code. The tradeoff resulting from the use of open source software came from the fact finding a well-organized and well documented process to enable the native capability of components was a major challenge, call this the open source challenge. The open source challenge was not restricted to the ground control software alone. The hardware components were built to be used by the open source community that developed and utilizes the ground control software, thus the documentation was equally challenging for the hardware as it was for the software. The community for the Mission Planner software was very active, constantly generating new capabilities and versions of the software. The Mission Planner community did not maintain the documentation at the same rate as the developments were released. Additionally, there were many users on the chat forums but getting a person to respond to a specific question was a challenge. The Mission Planner community was better than the QGroundControl community because the QGroundContol community was not very active. It was clear that there were some QGroundControl users still posting to forums but gaining account access to join in online discussions was unattainable for our research team.

As demonstrated by the success achieved for the majority of the objectives of this research effort, most of the open source challenge was overcome. The challenges that remain for developing any system based on the Ardupilot originate with the low maturity of the technology being applied. Obtaining a factual history of open source UAV technology is not a simple task. The open source UAV community contests the origins of some advances because the code is available for anybody to take, modify and introduce as their own. What is clear is that the commercial availability of lithium- polymer batteries in 1997 provided a dense and affordable power supply that attracted hobbyists and academic researchers to work with SUAS. The Ardupilot Mega has only been commercially available for less than three years. In the duration of this project a new version of the APM was released and the APM 2.0 utilized in this research effort was phased out of production. Additionally, there are multiple open source autopilot projects currently competing to be the leading platform in the autopilot community [16]. This creates a rapidly changing environment that causes some difficulty when trying to have a stable base to conduct independent research. There are multiple options to adapt to these open source challenges when looking forward to potential future work for this research area.

5.3 Future Work

Commercial SUAS are in their infancy today and advances keep coming with vibrant impetus. Any future work following on this thesis must be motivated by the requirements of the Department of Defense stake holders that fund this area of research because the future potential of applications is limitless. Given that as a preface, this research effort has inspired a few specific potential research projects that are divided into two categories: unlocking the potential of open source SUAS technology and developing new SUAS capabilities.

Unlocking the potential of open source SUAS technology presents risks, challenges and rewarding results. The open source SUAS community has not converged to any standard architecture or protocol. There are four established and active open source projects operating today: Ardupilot (DIY Drones), Paparazzi, OpenPilot, and PixHawk [13]. While all four open source projects developed from the same source—Paparazzi—enough differences exist between the projects to limit interchangeability of components and software across the platforms. MAVLink protocol has been introduced with the potential to increase the cross platform interchangeability of the SUAS open source communities.

Currently PixHawk firmware fully utilizes the MAVLink protocol. Ardupilot Mega firmware was developed prior to the release of MAVLink protocol and developed its own protocol; however, MAVLink protocol has been partially adapted by the DIY Drones Ardupilot Mega development community. If MAVLink protocol were fully implemented in Ardupilot Mega firmware and accepted by the open source development community, interchangeability between the two largest open source SUAS communities would be enabled. Ardupilot Mega users could fully utilize the ground control software and more advanced chipset of the PixHawk community. The PixHawk community would be more accessible to the established commercial and developer base of the Ardupilot Mega. Independent research efforts, like those pursued by the AFIT SUAS research team, would be able to draw on the capabilities of both the PixHawk and Ardupilot Mega communities.

The potential benefits do come with challenges and risks. The challenge of adapting the firmware for the Ardupilot to fully implement MAVLink protocol is not so much a technical research challenge but more closely described as a programming effort. Once programming is completed there is no guarantee that the Ardupilot Mega development managers will accept the new firmware. This would result in having a firmware developed for one generation of the Ardupilot Mega chipset that could operate with PixHawk software. Each time that chipset would be updated and the old chipset phased out the firmware would have to be tested for compatibility. Additionally, the ground control software developed for the original firmware of the Ardupilot Mega would no longer function with the full MAVLink enabled firmware. If the new firmware were accepted it would not immediately create any new capabilities. Both the PixHawk and the Ardupilot Mega are functional inside the scope of their similar, albeit independent, communities. The advantage gained would be left open to end users interested in capabilities developed in both open source communities.

Developing new SUAS capabilities has a wide range of possibilities. This research effort inspired two specific capabilities for development. An urban multi-rover relay SUAS could build on this research to provide an ISR platform capable of navigating in an urban environment. A relative proximity keeping SUAS could have a wide range of application from convoy security, to fully automated scouting, to parameter keeping. Both of these proposed capabilities would draw heavily on technology developed for the rover relay cooperative control SUAS.

An urban multi-rover relay SUAS could integrate 3D Google® mapping and aerial networking into ground control software to provide ISR capabilities in obstacle rittled environment. Open source ground control software developers have already released an alpha version of ground control software that integrates Google® 3D mapping into the flight planner. There would be a clear advantage to using multi-rotor UAVs in an urban environment because of their increased maneuverability and ability to hover as compared to fixed wing UAVs. The cooperative control autonomous algorithm developed for this research effort should be directly transferable to multi-rotor UAVs and the slower cruise speed combined with the ability to hover should yield more optimal flight trajectories compared to fixed wing UAVs. The objective would be to have a high altitude relay UAV autonomously position itself to relay communications to one or more rover UAVs operating at a lower altitude where buildings would obscure direct line-of-sight communications to a ground control station.

A relative proximity keeping SUAS could provide many Department of Defense related capabilities by modifying the rover relay cooperative control concept in a simple way. Instead of having a relay UAV that used the GPS data from a mobile rover UAV to autonomously navigate, the GPS could be attached to a ground station and the ground station could be mobilized. The autonomous navigation algorithm could be modified to allow the user to specify a relative position and/or trajectory from the ground control station to the UAV(s). As the ground control station moved so would the UAV(s). The changes to the autonomous navigation algorithm would be moderate yet could prove to have a wide range of applications. UAVs or unmanned ground vehicles escorts could travel with convoys to provide improvised explosive device screening and/or ISR capabilities. Units on patrol could launch UAV scouts and not have to provide any further navigation input as they proceeded on the patrol observing the scouts' ISR data. Mobile units of any kind could launch UAVs and have the UAVs perimeter keep without having to update the correct parameter position as the unit moved. Researchers have already demonstrated the ability of SUAS to track a target; however, being able to position a UAV arbitrarily relative to a moving ground reference has not been demonstrated.

5.4 Summary

The attempted end state objective of the research effort was to flight test an autonomous control algorithm on a relay UAV that was actively relaying data to a rover UAV in an extended effective line-of-sight operating range. Flight testing demonstrated a 94% success rate in developing the functionality necessary to achieve the end state objective. The "Vee" process model helped to keep the project in scope by focusing on the requirements needed to obtain the end state. Follow up research that came after the final flight testing has demonstrated solutions, during ground testing, to achieve 100% of the functional requirements to realize the end state objective. The future work building on the demonstrated technology developed in this research effort is expansive in it potential but comes with new challenges.

Appendix A. Test Procedures

Flight Test #1 Initial Flight Testing (24-25 September 2012)

1. Preflight testing
    a. Communication check (initial)
    b. Control Surface check
    c. Trim Radio and save settings
    d. Communication check (distance)
2. In Flight Testing With Mission Planner
    a. OWL_A1 & OWL_A2
        i. Zero Sensors
        ii. Set Fail Safe Parameters
        iii. Trim Radio
        iv. Load Waypoints
        v. Launch OWL_A*
        vi. RC Pilot Flight
            1. Adjust Trim
        vii. Engage Autopilot
            1. Adjust Gains (as necessary)
        viii. RC Pilot Landing
        ix. Group Discussion Observations
    b. Sig Rascal_P1 (Petrol) & Sig Rascal_E1 (Electric)
        i. Zero Sensors
        ii. Set Fail Safe Parameters
        iii. Trim Radio
        iv. Load Waypoints
        v. Launch Rascal_*
        vi. RC Pilot Flight
            1. Adjust Trim
        vii. Engage Autopilot
            1. Adjust Gains (as necessary)
        viii. RC Pilot Landing
        ix. Group Discussion Observations
3. In Flight Testing With QGroundControl
    a. Communication check (initial)
    b. Control Surface check
    c. OWL_A1 Flight
        i. Zero Sensors
        ii. Set Fail Safe Parameters
        iii. Trim Radio
        iv. Load Waypoints
        v. Launch OWL_A1
        vi. RC Pilot Flight To Elevation

            vii. Engage Autopilot (observe QGroundControl)
                1. Try update of race track in flight
            viii. Land OWL_A1
            ix. Group Discussion Observations
    d. OWL_A2 Flight
        i. Zero Sensors
        ii. Set Fail Safe Parameters
        iii. Trim Radio
        iv. Load Waypoints
        v. Launch OWL_A2
        vi. RC Pilot Flight To Elevation
        vii. Engage Autopilot
        viii. Land OWL_A2

4. Multi-Aircraft Simultaneous Flight 1 With QGroundControl
    a. Replace batteries in OWL_A1 & OWL_A2
    b. Zero Sensors in OWL_A1 & OWL_A2
    c. Set Fail Safe Parameters in OWL_A1 & OWL_A2
    d. Load Waypoints for OWL_A1(elevation 350ft) & OWL_A2 (elevation 200ft)
    e. Launch OWL_A1
    f. RC Pilot Flight To Elevation
    g. Engage Autopilot Observe Lap
    h. Launch OWL_A2
    i. RC Pilot Flight To Elevation
    j. Engage Autopilot Observe Lap
    k. Update Waypoints OWL_A1
    l. Update Waypoints OWL_A2
    m. Land OWL_A1
    n. Land OWL_A2
    o. Group Discussion Observations
5. Multi-Aircraft Simultaneous Flight 1 With QGroundControl
    a. Replace batteries in OWL_A1 & Refill Petrol in Sig Rascal_P1
    b. Zero Sensors in OWL_A1 & Sig Rascal_P1
    c. Set Fail Safe Parameters in OWL_A1 & Sig Rascal_P1
    d. Load Waypoints for OWL_A1(elevation 250ft) & Sig Rascal_P1 (elevation 400ft)
    e. Launch Sig Rascal_P1
    f. RC Pilot Flight To Elevation
    g. Engage Autopilot Observe Lap
    h. Launch OWL_A1
    i. RC Pilot Flight To Elevation
    j. Engage Autopilot Observe Lap
    k. Update Waypoints Sig Rascal_P1
    l. Update Waypoints OWL_A1

m. Land OWL_A1
n. Land Sig Rascal_P1
o. Group Discussion Observations

Flight Test #2 Full System Verification (5-7 November 2012)

1. Initial communications check out
    a. Video feed check (5.4 GHz)
        i. Initial Operation
            1. Is Video feed working?
    b. RC Safety Pilot check (2.4 GHz)
        i. Initial Operation
            1. Is RC Communications working?
        ii. Distance check
            1. On the ground place the FrSky transmitter in range check mode and walk the MAV down the flight line until communications are lost. Do conversion for approximated RC range. Record here __________________
    c. Auto Pilot check (914 MHz)
        i. Initial Operation
            1. Is RC Communications working?
        ii. Distance check
            1. Walk the MAV down the flight line until communications are lost. Record distance here __________________
    d. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance
2. Verify MAVs are flying properly (In Flight Testing With Mission Planner)
    a. Power on RC controllers for OWL_A1 and OWL_A2
    b. For Each OWL_A1, OWL_A2 and Sig_AP
        i. Open Mission Planner
        ii. **Connec**t to MAV at baud rate of 57600
        iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        iv. Verify that the altitude read out on the right of the flight data screen reads **0**
        v. **Repeat** iii-iv as necessary until successful
        vi. Trim Radio
        vii. Load Waypoints
        viii. Launch MAV
        ix. RC Pilot Flight
            1. Adjust Trim

            x. Engage Autopilot
                1. Adjust Gains (as necessary) **SEE APPENDIX B**
            xi. RC Pilot Landing
        c. Group Discussion Observations
        d. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance

3. Single MAV flight using QGroundControl (First test OWL_A2 , repeat procedure for Sig_AP)
    a. Power on RC controllers OWL_A2 and Sig_AP
    b. Zero Sensors
        i. **Open Mission Planner**
        ii. **Connec**t to MAV at baud rate of 57600
        iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        iv. Verify that the altitude read out on the right of the flight data screen reads **0**
        v. **Repeat** as necessary until successful
        vi. **Close Mission Planner but do NOT power off MAV**
    c. Trim Radio
    d. **Open UNMODIFIED qgroundcontrol**
    e. **Connect** to MAV at baud rate of 57600
    f. **Wait for GPS to find location**
    g. **Load Waypoints** using waypoint widget
    h. **Verify Waypoints** by going to the onboard tab of the waypoint widget and clicking refresh
    i. Launch
    j. RC Pilot Flight To Elevation
    k. Engage Autopilot
        i. Try update of race track in flight
        ii. Observe data logging capabilities
    l. Land
    m. Group Discussion Observations
    n. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance

4. Single MAV Distance Flight to Loss of Communications
    a. Power on RC controllers for OWL_A2
    b. Zero Sensors
        i. **Open Mission Planner**
        ii. **Connec**t to OWL_A2 at baud rate of 57600
        iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        iv. Verify that the altitude read out on the right of the flight data screen reads **0**
        v. **Repeat** as necessary until successful

c. Trim Radio
d. **Wait for GPS to find location**
e. **Load Waypoints** using waypoint widget
f. **Verify Waypoints** by going to the onboard tab of the waypoint widget and clicking refresh
g. Send Safety pilot and Observers to remote location (Must have range radio)
    i. Observer will have map of flight pattern
h. **Verify both teams are ready and we are clear for launch**
i. Launch
j. RC Pilot Flight To Elevation
k. RC Pilot flies OWL_A2 toward primary ground station
l. Ground control operator is continually attempting to connect
m. Monitor telemetry to observe when 914 MHz communications are established
n. Ground control operator notes distance on map where communications were established
o. Observe if after 30 seconds of flight OWL_A2 beings to navigate toward RTL
p. Operator then notifies RC pilot to land OWL_A2
q. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance

5. Multi-MAV Multi-Ground Station Familiarity Test (Direct LOS) Non-autonomous Relay Navigation
    a. Power on RC controllers for OWL_A1 and OWL_A2
    b. On two separate Laptops connect two Digi modems (one to each laptop)
    c. Open X-CTU and verify that each computer is talking to the attached modem successfully
        i. Select the test/query button. The computer is successfully connected if the type and model information is not garbled text
    d. On laptop one (L1) open Mission Planner
        i. **Power on** OWL_A1 while **holding the MAV level and steady**
        ii. **Connec**t to OWL_A1 at baud rate of 57600
        iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        iv. Verify that the altitude read out on the right of the flight data screen reads **0**
        v. **Repeat** iii-iv as necessary until successful
        vi. Trim Radio
        vii. Load Waypoints
    e. On laptop two (L2) open Mission Planner
        i. Zero Sensors

1. **Open Mission Planner**
2. **Connec**t to OWL_A2 at baud rate of 57600
3. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
4. Verify that the altitude read out on the right of the flight data screen reads **0**
5. **Repeat** as necessary until successful
6. **Close Mission Planner but do NOT power off MAV**

ii. Trim Radio
iii. **Open UNMODIFIED qgroundcontrol**
iv. **Connect** to MAV at baud rate of 57600
v. **Wait for GPS to find location**
vi. **Load Waypoints** using waypoint widget
vii. **Verify Waypoints** by going to the onboard tab of the waypoint widget and clicking refresh

f. Launch OWL_A1
   i. RC Pilot Flight To Elevation
   ii. Engage Autopilot
   iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else
g. Launch OWL_A2
   i. RC Pilot Flight To Elevation
   ii. Engage Autopilot
   iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else
h. Maximize flight time of OWL_A1 to 15 minutes of flight without exceeding time limit
i. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance

6. Multi-MAV Multi-Ground Station Familiarity Test (Direct LOS) Autonomous Relay Navigation
   a. Power on RC controllers for OWL_A1 and OWL_A2
   b. On two separate Laptops connect two Digi modems (one to each laptop)
   c. Open X-CTU and verify that the computer is talking to the modem successfully
      i. Select the test/query button. The computer is successfully connected if the type and model information is not garbled text
   d. On laptop one (L1) open Mission Planner
      i. **Power on** OWL_A1 while **holding the MAV level and steady**
      ii. **Connec**t to OWL_A1 at baud rate of 57600
      iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**

iv. Verify that the altitude read out on the right of the flight data screen reads **0**
v. **Repeat** iii-iv as necessary until successful
vi. Trim Radio
vii. Load Waypoints at altitude of 550 ft

e. On laptop two (L2) open Mission Planner
i. Zero Sensors
1. **Open Mission Planner**
2. **Connec**t to OWL_A2 at baud rate of 57600
3. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
4. Verify that the altitude read out on the right of the flight data screen reads **0**
5. **Repeat** as necessary until successful
6. **Close Mission Planner but do NOT power off OWL_A2**

ii. Trim Radio
iii. **Open MODIFIED qgroundcontrol**
iv. **Connect** to both MAVs at baud rate of 57600 (do not enable multiplexing)
v. **Wait for GPS to find location**
vi. **Click** on map as close as possible to the location of the ground station as possible

f. Launch OWL_A1
i. RC Pilot Flight To Elevation
ii. Engage Autopilot
iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else

g. Launch OWL_A2
i. RC Pilot Flight To Elevation
ii. Engage Autopilot
iii. Every 5 seconds click anywhere on the map
iv. Verify Operation Status (if oddities are observed, land and trouble shoot) else

h. Maximize flight time of first MAV to 15 minutes of flight without exceeding time limit
i. Take manual control of MAV OWL_A2 and land it
ii. Take manual control of MAV OWL_A1 and land it

i. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance

7. Multi-MAV Multi-Ground Station Familiarity Test (Direct LOS) Autonomous Relay Navigation **with SIG_AP in place of OWL_A2**
a. Power on RC controllers for OWL_A1 and OWL_A2
b. Switch Sig_AP Aircraft ON (leave Autopilot switch OFF)

c. **Power on** OWL_A1 while **holding the MAV level and steady**
d. On laptop one (L1) open Mission Planner
    i. Plug in Ch1-Relay modem to laptop L1
    ii. **Connec**t to OWL_A1 at baud rate of 57600
    iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
    iv. Verify that the altitude read out on the right of the flight data screen reads **0**
    v. **Repeat** iii-iv as necessary until successful
    vi. Trim Radio
    vii. Load Waypoints
e. Switch Sig_AP Autopilot ON
f. On laptop two (L2) open Mission Planner
    i. Plug in Ch1-Sig modem to laptop L2
    ii. Zero Sensors
        1. Open Mission Planner
        2. **Connec**t to Sig_AP at baud rate of 57600
        3. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        4. Verify that the altitude read out on the right of the flight data screen reads **0**
        5. **Repeat** as necessary until successful
        6. Hold Sig_AP level
        7. Under the configuration tab click on the calibrate level
        8. Verify on the flight data tab that the hud is showing level flight
        9. **Close Mission Planner but do NOT power off MAV**
    iii. Trim Radio
    iv. **Open MODIFIED qgroundcontrol**
    v. **Connect** to Sig_AP at baud rate of 57600
    vi. **Wait for GPS to find location**
    vii. **Select** MAV001 (Sig) for control
    viii. **Load Waypoints** using waypoint widget
    ix. **Verify Waypoints** by going to the onboard tab of the waypoint widget and clicking refresh
g. Launch OWL_A1
    i. RC Pilot Flight To Elevation
    ii. Engage Autopilot
    iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else
h. Launch Sig_AP
    i. RC Pilot Flight To Elevation
    ii. Engage Autopilot

iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else

i. Maximize flight time of OWL_A1 to 15 minutes of flight without exceeding time limit

   i. Take manual control of MAV Sig_AP and land it
   ii. Take manual control of MAV OWL_A1 and land it

j. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance

8. Beyond Communications Line-of-sight (BCLOS) Flight Test
   a. Power on RC controllers for OWL_A1 and OWL_A2
   b. Switch Sig_AP Aircraft ON (leave Autopilot switch OFF)
   c. **Power on** OWL_A1 while **holding the MAV level and steady**
   d. On laptop one (L1) open Mission Planner
      i. Plug in Ch1-Relay modem to laptop L1
      ii. **Connec**t to OWL_A1 at baud rate of 57600
      iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
      iv. Verify that the altitude read out on the right of the flight data screen reads **0**
      v. **Repeat** iii-iv as necessary until successful
      vi. Trim Radio
      vii. Load Waypoints
   e. Switch Sig_AP Autopilot ON
   f. On laptop two (L2) open Mission Planner
      i. Plug in Ch1-Sig modem to laptop L2
      ii. Zero Sensors
         1. **Open Mission Planner**
         2. **Connec**t to Sig_AP at baud rate of 57600
         3. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
         4. Verify that the altitude read out on the right of the flight data screen reads **0**
         5. **Repeat** as necessary until successful
         6. Hold Sig_AP level
         7. Under the configuration tab click on the calibrate level
         8. Verify on the flight data tab that the hud is showing level flight
         9. **Close Mission Planner but do NOT power off MAV**
      iii. Trim Radio
      iv. **Open MODIFIED qgroundcontrol**
      v. **Connect** to Sig_AP at baud rate of 57600
      vi. **Wait for GPS to find location**

vii. **Select** MAV001 (Sig) for control
viii. **Load Waypoints** using waypoint widget
ix. **Verify Waypoints** by going to the onboard tab of the waypoint widget and clicking refresh

g. Send out RC pilot and distant area observer with map of flight path, cell phone and range radio
h. Launch SIG_AP
    i. RC Pilot Flight To Elevation and approximate relay position
i. Launch OWL_A1
    i. RC Pilot Flight To Elevation
    ii. Engage Autopilot
    iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else
j. Ground Control Operator verifies that relay of communications is operational
    i. Is telemetry data displaying in the ground control software?
    ii. Can information be written to the rover MAV?
    iii. If yes proceed. If no fly OWL_A1 closer to Sig_AP.
k. **On Sig_AP**
    i. Engage Autopilot
    ii. Every 5 seconds click anywhere on the map
    iii. Verify Operation Status (if oddities are observed, land and trouble shoot)
l. Maximize flight time of OWL_A1 to 15 minutes of flight without exceeding time limit
m. On ground control operator's que both RC pilots take control of their respective MAVs and land the MAVs
n. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance

9. Stationary Target Flight Test
    a. Emplace stationary target
    b. Set waypoint pattern to loiter over target
    c. Launch OWL and monitor to ensure proper flight path
    d. Record and Measure loiter time and target observed time
10. Road Surveillance Flight Test
    a. Designate linear zone of observation
    b. Set waypoint pattern to observe linear zone of observation
    c. Launch OWL and monitor to ensure proper flight path
    d. Record and Measure loiter time and target observed time

Appendix B. Gain Tuning Procedures

Ardupilot Gain Tuning Guide

Author: Charles Neal

Note 1: This guide is designed to tune the gains for an aircraft that has already been setup in an approved configuration in which the RC transmitter is functioning correctly, all desired autopilot settings and fail safes have been verified, mode selection works properly, and has already been flown in RC mode to ensure trim settings and flight characteristics are acceptable. This document is meant as a user guide and any appropriate test plans and/or safety appendices should be followed. Both users (RC pilot and ground station operator) should be familiar with the Ardupilot system and in compliance with any current proficiency requirements.

Note 2: It is expected that this process will require multiple flights. Before the power on the ground station is ever cycled, if any aircraft configure file changes have been made (to include all gains), the configure file must be saved.

Step	A/P Mode	Action	Response	Notes
1	Stabilize	Check servo response directions: On ground, manually induce pitch, roll, and yaw. Ensure servo response opposes motion.	If all directions are good: Step 2; If directions reversed: Check appropriate reversing boxes in aircraft control surface configure tab and repeat step 1.	Beginning with Step 1, ensure that all three feed-forward mix gains (rudder mix, P-to-T, PitchComp) are either set to zero or left at the low default value.
2	N/A	Set desired bank and pitch limits in aircraft configuration tab. Click "Write Params" when done (updates AP flash).	Continue to step 3.	For the remainder of the process "write params" must be used to store any values that are updated in the aircraft configuration file.
3A	Stabilize	Set Servo_Roll P Gain: In flight, switch to stabilize mode and observe aircraft roll (bump stick to induce disturbances). RC pilot should observe aircraft and GSO can view real-time chart of aircraft attitude and servo responses.	If under damped (excessive oscillation, overshoot, and/or increasing amplitude): reduce servo_roll P gain; If over damped (insufficient response): increase gain; If aircraft stabilizes adequately: Step 3B.	Once an adequate gain is found, a decent rule of thumb is to slowly increase the gain until oscillation is first visible, then use 50%-75% of that value. That will generally result in a decent gain with enough of a margin of error to avoid being adversely affected by minor changes.

3B	Stabilize	Set Servo_Pitch P Gain: In flight, switch to stabilize mode and observe aircraft pitch (bump stick to induce disturbances). RC pilot should observe aircraft and GSO can view real-time chart of aircraft attitude and servo responses.	If under damped: reduce servo_pitch P gain; If over damped: increase gain; If aircraft stabilizes adequately: Step 3C.	See 3A notes.
3C	Stabilize	Set Servo_Yaw P Gain (if yaw dampening): In flight, switch to stabilize mode and observe aircraft yaw (bump stick to induce disturbances). RC pilot should observe aircraft and GSO can view real-time chart of aircraft attitude and servo responses.	If under damped: reduce servo_yaw P gain; If over damped: increase gain; If aircraft stabilizes adequately: Step 4 or return to step 3A (see note).	Step 3 is iterative: If initial gain in any one flight axis is causing severe aircraft behavior while attempting to tune a different axis servo gain, it may be necessary to cycle through step 3 multiple times until stabilize mode is adequate.
4	N/A	Set approximate throttle settings (on ground): In aircraft configuration tab, enter throttle min, max, and approximate cruise setting to be used in initial navigation gain tuning.	Continue to step 5.	Incorrect cruise throttle setting may result in throttle oscillation. For this step use a conservative initial estimate. If throttle oscillation is still present after completing step 6B, adjust cruise throttle setting in small increments in a direction that reduces oscillation.
5	Autonomous	Set Nav_Roll P gain: In flight, switch to autonomous mode and observe aircraft heading while attempting to maintain a racetrack pattern. RC pilot should observe aircraft and GSO can view real-time chart of aircraft heading.	If heading is under damped: reduce Nav_roll P gain; If heading is over damped: increase gain; If heading tracking is adequate: step 6.	Ensure crosstrack is turned off (gain=0) while completing step 5.

6A	Autonomous	Set Nav_PitchAS P gain: In flight, observe airspeed during straight and level flight. Induce changes by commanding increases and decreases in airspeed. Observe pitch behavior of aircraft.	If under damped: reduce nav_pitchAS P gain; If response is over damped: increase gain; If aircraft adequately attains a target airspeed: Step 6B.	One method to reduce coupling with Energy/Altitude performance is to command a large altitude change and tune Nav_PitchAS while climbing or descending. This should command throttle to max or min setting and isolate airspeed control to pitch.
6B	Autonomous	Set Energy/Alt P gain: In flight, observe altitude during straight and level flight, Induce changes by commanding increases and decreases in altitude. Observe throttle behavior of aircraft.	If throttle and/or altitude oscillation occurs (under damped): reduce Energy/Alt P gain; If response is over damped: increase gain; If aircraft reaches and maintains target altitude adequately: Step 7 or return to step 6A (see note).	The behavior of Nav_PitchAS and Energy/Alt are coupled. It may be necessary to cycle through step 6 multiple times until both airspeed and altitude changes without tuning required to either P gain. Regarding Energy/Alt: if altitude is holding acceptably but throttle oscillation is still observed, see note on step 4.
7	Autonomous	Activate pitch to throttle mix if required/desired (will reduce inadvertent altitude coupling with airspeed changes): Increase P-to-T gain (should be 0 initially) and, starting from straight and level flight, command both increases and decreases in airspeed.	If immediate coupling between airspeed changes and altitude is high: increase gain by small amount; If coupling is minimal/acceptable: Step 8.	N/A
8A	Fly By Wire	Set PitchComp: In FBW mode, start from straight and level flight and command full bank. Observe immediate pitch behavior of aircraft.	If aircraft immediately pitches down: increase PitchComp gain; If aircraft immediately pitches up: decrease gain; If aircraft maintains pitch well while banking: Step 8B.	This is the most direct method for tuning PitchComp, however this may also be tuned in autonomous mode. While flying a basic racetrack pattern, observe pitch behavior whenever waypoint changes result in a transition from

				straight and level flight to a bank.
8B	Fly By Wire	Set Rudder Mix: In FBW mode, start from straight and level flight and command full bank. Observe immediate turn coordination of aircraft.	If aircraft is initially uncoordinated in turn (adverse yaw): increase Rudder Mix gain; If aircraft immediately overshoots a coordinated yaw attitude: decrease gain; If aircraft initially coordinates turn well: Step 9.	This is the most direct method for tuning Rudder Mix, however this may also be tuned in autonomous mode. While flying a basic racetrack pattern, observe turn coordination whenever waypoint changes result in a transition from straight and level flight to a bank
9	Autonomous	Set Cross Track Settings: Set desired Xtrack Entry Angle and initial gain. In autonomous flight, command a racetrack pattern. When waypoint changes occur, observe ability of aircraft to return to ideal path and maintain desired entry angle.	If aircraft oscillates about desired entry path and waypoint path: decrease crosstrack gain; If aircraft does not achieve entry angle or waypoint path: increase gain; If crosstrack behavior is acceptable: Done.	N/A

Appendix C. Advanced Parameter Settings

Sig-Rascal 110 Advanced Parameters List

This list is the all parameter settings used for well-adjusted autopilot flight of the Sig-Rascal.

AHRS_BARO_USE	0		0:Disabled,1: Enabled	This controls the use of the barometer for vertical acceleration compensation in AHRS. It is currently recommended that you set this value to zero unless you are a developer experimenting with the AHRS system.
AHRS_GPS_GAIN	1		0.0 1.0	This controls how much to use the GPS to correct the attitude. This should never be set to zero for a plane as it would result in the plane losing control in turns. For a plane please use the default value of 1.0.
AHRS_GPS_USE	1			This controls whether to use dead-reckoning or GPS based navigation. If set to 0 then the GPS won't be used for navigation, and only dead reckoning will be used. A value of zero should never be used for normal flight.
AHRS_RP_P	0.4		0.1 0.4	This controls how fast the accelerometers correct the attitude
AHRS_WIND_MAX	0		0 127	This sets the maximum allowable difference between ground speed and airspeed. This allows the plane to cope with a failing airspeed sensor. A value of zero means to use the airspeed as is.
AHRS_YAW_P	0.4		0.1 0.4	This controls the weight the compass or GPS has on the heading. A higher value means the heading will track the yaw source (GPS or compass) more rapidly.
ALT_CTRL_ALG	0		0:Default Method,1:non-airspeed	This sets what algorithm will be used for altitude control. The default is to select the algorithm based on whether airspeed is enabled. If you set it to 1, then the airspeed based algorithm won't be used for altitude control, but airspeed can be used for other flight control functions
ALT_HOLD_FBW	0			

CM				
ALT_HOLD_RTL	15000	centimeters		Return to launch target altitude
ALT_MIX	1	Percent	0 1	The percent of mixing between gps altitude and baro altitude. 0 = 100% gps, 1 = 100% baro
ALT_OFFSET	0	Meters	-32767 32767	This is added to the target altitude in automatic flight. It can be used to add a global altitude offset to a mission, or to adjust for barometric pressure changes
ALT2PTCH_D	0			
ALT2PTCH_I	0.1			
ALT2PTCH_IMAX	500			
ALT2PTCH_P	0.65			
AMP_PER_VOLT	27.32			
ARSP2PTCH_D	0			
ARSP2PTCH_I	0.3			
ARSP2PTCH_IMAX	500			
ARSP2PTCH_P	0.9			
ARSPD_ENABLE	1		0:Disable,1:Enable	enable airspeed sensor
ARSPD_FBW_MAX	22	m/s	5 50	Airspeed corresponding to maximum throttle in Fly By Wire B mode.
ARSPD_FBW_MIN	6	m/s	5 50	Airspeed corresponding to minimum throttle in Fly By Wire B mode.
ARSPD_OFFSET	596.681			Airspeed calibration offset
ARSPD_RATIO	1.994			Airspeed calibration ratio
ARSPD_USE	0		1:Use,0:Don't Use	use airspeed for flight control
BATT_CAPACITY	1760	mAh		Capacity of the battery in mAh when full
BATT_MONITOR	0			
CAM_TRIGG_TYPE	0		0:Servo,1:Relay,2:Servo and turn off throttle,3:Ser	how to trigger the camera to take a picture

			vo when 3m from waypoint,4:transistor	
CMD_INDEX	0			
CMD_TOTAL	2			
COMPASS_AUTODEC?	1			
COMPASS_DEC	0			
COMPASS_LEARN	1			
COMPASS_OFS_X	-17.351			
COMPASS_OFS_Y	32.892			
COMPASS_OFS_Z	-10.953			
COMPASS_USE	1			
ELEVON_CH1_REV	0		-1:Disabled,1:Enabled	Reverse elevon channel 1
ELEVON_CH2_REV	0		-1:Disabled,1:Enabled	Reverse elevon channel 2
ELEVON_MIXING	0			
ELEVON_REVERSE	0		0:Disabled,1:Enabled	Reverse elevon mixing
ENRGY2THR_D	0			
ENRGY2THR_I	0.35			
ENRGY2THR_IMAX	400			
ENRGY2THR_P	0.75			
FBWB_ELEV_REV	0		0:Disabled,1:Enabled	Reverse sense of elevator in FBWB. When set to 0 up elevator (pulling back on the stick) means to lower altitude. When set to 1, up elevator means to raise altitude.
FENCE_ACTION	0		0:None,1:Gui	What to do on fence breach

			dedMode,2:ReportOnly	
FENCE_CHANNEL	0			RC Channel to use to enable geofence. PWM input above 1750 enables the geofence
FENCE_MAXALT	0	meters	0 32767	Maximum altitude allowed before geofence triggers
FENCE_MINALT	0	meters	0 32767	Minimum altitude allowed before geofence triggers
FENCE_TOTAL	0			Number of geofence points currently loaded
FLAP_1_PERCNT	0			
FLAP_1_SPEED	0			
FLAP_2_PERCNT	0			
FLAP_2_SPEED	0			
FLTMODE_CH	8			RC Channel to use for flight mode control
FLTMODE1	11		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Loiter,15:Guided	Flight mode for switch position 1 (910 to 1230 and above 2049)
FLTMODE2	11		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Loiter,15:Guided	Flight mode for switch position 2 (1231 to 1360)
FLTMODE3	10		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Loiter,15:Guided	Flight mode for switch position 3 (1361 to 1490)
FLTMODE4	10		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Loiter,15:Guided	Flight mode for switch position 4 (1491 to 1620)

FLTMODE5	2		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Loiter,15:Guided	Flight mode for switch position 5 (1621 to 1749)
FLTMODE6	0		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Loiter,15:Guided	Flight mode for switch position 6 (1750 to 2049)
FORMAT_VERSION	13			
FS_GCS_ENABL	0		0:Disabled,1:Enabled	Enable ground control station telemetry failsafe. Failsafe will trigger after 20 seconds of no MAVLink heartbeat messages
FS_LONG_ACTN	0		0:None,1:ReturnToLaunch	The action to take on a long (20 second) failsafe event
FS_SHORT_ACTN	0		0:None,1:ReturnToLaunch	The action to take on a short (1 second) failsafe event
GND_ABS_PRESS	99785.3			
GND_TEMP	23.445			
HDNG2RLL_D	0.1			
HDNG2RLL_I	0.1			
HDNG2RLL_IMAX	500			
HDNG2RLL_P	1.2			
IMU_PRODUCT_ID	0			
INPUT_VOLTS	4.68			
INVERTEDFLT_CH	0			
KFF_PTCH2THR	0.1		0 5	Pitch to throttle feed-forward gain.
KFF_PTCHCOMP	0.2		0 1	Adds pitch input to compensate for the loss of lift due to roll control. 0 = 0 %, 1 = 100%
KFF_RDDRMIX	0.5		0 1	The amount of rudder mix to apply during aileron movement 0 = 0 %, 1 = 100%

KFF_THR2PTCH	0		0 5	Throttle to pitch feed-forward gain.
LAND_FLARE_ALT	3			
LAND_FLARE_SEC	2			
LAND_PITCH_CD	0			
LIM_PITCH_MAX	2000	centi-Degrees	0 9000	The maximum commanded pitch up angle
LIM_PITCH_MIN	-2000	centi-Degrees	-9000 0	The minimum commanded pitch down angle
LIM_ROLL_CD	4500	centi-Degrees	0 9000	The maximum commanded bank angle in either direction
LOG_BITMASK	334			bitmap of log fields to enable
MAG_ENABLE	1		0:Disabled,1:Enabled	Setting this to Enabled(1) will enable the compass. Setting this to Disabled(0) will disable the compass
MANUAL_LEVEL	0		0:Disabled,1:Enabled	Setting this to Disabled(0) will enable autolevel on every boot. Setting it to Enabled(1) will do a calibration only when you tell it to
MIN_GNDSPD_CM	0	cm/s		Minimum ground speed in cm/s when under airspeed control
MNT_ANGMAX_PAN	4500	centi-Degrees	-18000 17999	Maximum physical pan (yaw) angular position of the mount
MNT_ANGMAX_ROL	4500	centi-Degrees	-18000 17999	Maximum physical roll angular position of the mount
MNT_ANGMAX_TIL	4500	centi-Degrees	-18000 17999	Maximum physical tilt (pitch) angular position of the mount
MNT_ANGMIN_PAN	-4500	centi-Degrees	-18000 17999	Minimum physical pan (yaw) angular position of mount.
MNT_ANGMIN_ROL	-4500	centi-Degrees	-18000 17999	Minimum physical roll angular position of mount.
MNT_ANGMIN_TIL	-4500	centi-Degrees	-18000 17999	Minimum physical tilt (pitch) angular position of mount.

MNT_CONTROL_X	0			
MNT_CONTROL_Y	0			
MNT_CONTROL_Z	0			
MNT_JSTICK_SPD	0		0 10	0 for position control, small for low speeds, 10 for max speed
MNT_MODE	0		0:retract,1:neutral,2:MavLink_targeting,3:RC_targeting,4:GPS_point	Camera or antenna mount operation mode
MNT_NEUTRAL_X	0			
MNT_NEUTRAL_Y	0			
MNT_NEUTRAL_Z	0			
MNT_RC_IN_PAN	0		0:Disabled,5:RC5,6:RC6,7:RC7,8:RC8	0 for none, any other for the RC channel to be used to control pan (yaw) movements
MNT_RC_IN_ROLL	0		0:Disabled,5:RC5,6:RC6,7:RC7,8:RC8	0 for none, any other for the RC channel to be used to control roll movements
MNT_RC_IN_TILT	0		0:Disabled,5:RC5,6:RC6,7:RC7,8:RC8	0 for none, any other for the RC channel to be used to control tilt (pitch) movements
MNT_RETRACT_X	0			
MNT_RETRACT_Y	0			
MNT_RETRACT_Z	0			
MNT_STAB_PAN	0		0:Disabled,1:Enabled	enable pan (yaw) stabilization relative to Earth
MNT_STAB_ROLL	0		0:Disabled,1:Enabled	enable roll stabilization relative to Earth
MNT_STAB_TILT	0		0:Disabled,1:Enabled	enable tilt (pitch) stabilization relative to Earth
PTCH2SRV_D	0			
PTCH2SRV_I	0.25			
PTCH2SRV_IMA	50			

X	0			
PTCH2SRV_P	1.1			
RC1_DZ	30			dead zone around trim.
RC1_MAX	2016	ms	800 2200	RC maximum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC1_MIN	998	ms	800 2200	RC minimum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC1_REV	1		-1:Reversed,1:Normal	Reverse servo operation. Ignored on APM1 unless dip-switches are disabled.
RC1_TRIM	1200	ms	800 2200	RC trim (neutral) PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC10_DZ	0			
RC10_FUNCTION	0		0:Disabled,1:Manual,2:Flap,3:Flap_auto,4:Aileron,5:flaperon,6:mount_pan,7:mount_tilt,8:mount_roll,9:mount_open,10:camera_trigger,11:release,12:mount2_pan,13:mount2_tilt,14:mount2_roll,15:mount2_open,16:DifferentialSpoiler1,17:DifferentialSpoiler2,18:AileronWithInput	Setting this to Disabled(0) will disable this output, any other value will enable the corresponding function
RC10_MAX	1900			
RC10_MIN	1100			
RC10_REV	1			
RC10_TRIM	1500			

RC11_DZ	0			
RC11_FUNCTION	0		0:Disabled,1:Manual,2:Flap,3:Flap_auto,4:Aileron,5:flaperon,6:mount_pan,7:mount_tilt,8:mount_roll,9:mount_open,10:camera_trigger,11:release,12:mount2_pan,13:mount2_tilt,14:mount2_roll,15:mount2_open,16:DifferentialSpoiler1,17:DifferentialSpoiler2,18:AileronWithInput	Setting this to Disabled(0) will disable this output, any other value will enable the corresponding function
RC11_MAX	1900			
RC11_MIN	1100			
RC11_REV	1			
RC11_TRIM	1500			
RC2_DZ	30			dead zone around trim.
RC2_MAX	2017	ms	800 2200	RC maximum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC2_MIN	1001	ms	800 2200	RC minimum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC2_REV	1		-1:Reversed,1:Normal	Reverse servo operation. Ignored on APM1 unless dip-switches are disabled.
RC2_TRIM	1200	ms	800 2200	RC trim (neutral) PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC3_DZ	3			dead zone around trim.
RC3_MAX	18	ms	800 2200	RC maximum PWM pulse width. Typically

	98			1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC3_MIN	99 0	ms	800 2200	RC minimum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC3_REV	-1		-1:Reversed,1: Normal	Reverse servo operation. Ignored on APM1 unless dip-switches are disabled.
RC3_TRIM	18 92	ms	800 2200	RC trim (neutral) PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC4_DZ	30			dead zone around trim.
RC4_MAX	20 16	ms	800 2200	RC maximum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC4_MIN	99 2	ms	800 2200	RC minimum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC4_REV	-1		-1:Reversed,1: Normal	Reverse servo operation. Ignored on APM1 unless dip-switches are disabled.
RC4_TRIM	12 00	ms	800 2200	RC trim (neutral) PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC5_DZ	0			
RC5_FUNCTION	0		0:Disabled,1: Manual,2:Flap,3:Flap_auto,4:Aileron,5: flaperon,6:mount_pan,7: mount_tilt,8: mount_roll,9 :mount_open ,10:camera_trigger,11:release,12:mount2_pan,13:mount2_tilt,14 :mount2_roll ,15:mount2_open,16:DifferentialSpoiler 1,17:DifferentialSpoiler2,1	Setting this to Disabled(0) will disable this output, any other value will enable the corresponding function

			8:AileronWithInput	
RC5_MAX	1554			
RC5_MIN	1553			
RC5_REV	1			
RC5_TRIM	1554			
RC6_DZ	0			
RC6_FUNCTION	0		0:Disabled,1:Manual,2:Flap,3:Flap_auto,4:Aileron,5:flaperon,6:mount_pan,7:mount_tilt,8:mount_roll,9:mount_open,10:camera_trigger,11:release,12:mount2_pan,13:mount2_tilt,14:mount2_roll,15:mount2_open,16:DifferentialSpoiler1,17:DifferentialSpoiler2,18:AileronWithInput	Setting this to Disabled(0) will disable this output, any other value will enable the corresponding function
RC6_MAX	1499			
RC6_MIN	1498			
RC6_REV	1			
RC6_TRIM	14			

	99			
RC7_DZ	0			
RC7_FUNCTION	0		0:Disabled,1:Manual,2:Flap,3:Flap_auto,4:Aileron,5:flaperon,6:mount_pan,7:mount_tilt,8:mount_roll,9:mount_open,10:camera_trigger,11:release,12:mount2_pan,13:mount2_tilt,14:mount2_roll,15:mount2_open,16:DifferentialSpoiler1,17:DifferentialSpoiler2,18:AileronWithInput	Setting this to Disabled(0) will disable this output, any other value will enable the corresponding function
RC7_MAX	1499			
RC7_MIN	1498			
RC7_REV	1			
RC7_TRIM	1499			
RC8_DZ	0			
RC8_FUNCTION	0		0:Disabled,1:Manual,2:Flap,3:Flap_auto,4:Aileron,5:flaperon,6:mount_pan,7:mount_tilt,8:mount_roll,9:mount_open,10:camera_trigger,11:release,12:mount2_pan,13:m	Setting this to Disabled(0) will disable this output, any other value will enable the corresponding function

			ount2_tilt,14:mount2_roll,15:mount2_open,16:DifferentialSpoiler1,17:DifferentialSpoiler2,18:AileronWithInput	
RC8_MAX	1863			
RC8_MIN	990			
RC8_REV	1			
RC8_TRIM	1605			
RC9_DZ	0			
RC9_FUNCTION	0		0:Disabled,1:Manual,2:Flap,3:Flap_auto,4:Aileron,5:flaperon,6:mount_pan,7:mount_tilt,8:mount_roll,9:mount_open,10:camera_trigger,11:release,12:mount2_pan,13:mount2_tilt,14:mount2_roll,15:mount2_open,16:DifferentialSpoiler1,17:DifferentialSpoiler2,18:AileronWithInput	Setting this to Disabled(0) will disable this output, any other value will enable the corresponding function
RC9_MAX	1900			
RC9_MIN	1100			
RC9_REV	1			
RC9_TRIM	15			

	00			
RLL2SRV_D	0			
RLL2SRV_I	0.2			
RLL2SRV_IMAX	50 0			
RLL2SRV_P	1			
RST_MISSION_C H	0			RC channel to use to reset the mission to the first waypoint. When this channel goes above 1750 the mission is reset. Set RST_MISSION_CH to 0 to disable.
RST_SWITCH_C H	0			RC channel to use to reset to last flight mode after geofence takeover.
RUDDER_STEER	0		0:Disabled,1: Enabled	When enabled, only rudder will be used for steering during takeoff and landing, with the ailerons used to hold the plane level
SCALING_SPEED	15	m/s		Airspeed in m/s to use when calculating surface speed scaling. Note that changing this value will affect all PID values
SERIAL3_BAUD	57		1:1200,2:240 0,4:4800,9:9 600,19:1920 0,38:38400,5 7:57600,111: 111100,115: 115200	The baud rate used on the telemetry port
SR0_EXT_STAT	2			
SR0_EXTRA1	10			
SR0_EXTRA2	10			
SR0_EXTRA3	2			
SR0_PARAMS	50			
SR0_POSITION	3			
SR0_RAW_CTRL	0			
SR0_RAW_SENS	2			
SR0_RC_CHAN	2			
SR3_EXT_STAT	0			
SR3_EXTRA1	0			
SR3_EXTRA2	0			
SR3_EXTRA3	0			
SR3_PARAMS	0			
SR3_POSITION	0			
SR3_RAW_CTRL	0			
SR3_RAW_SENS	0			
SR3_RC_CHAN	0			

STICK_MIXING	1		0:Disabled,1: Enabled	When enabled, this adds user stick input to the control surfaces in auto modes, allowing the user to have some degree of flight control without changing modes
SYS_NUM_RESETS	137			Number of APM board resets
SYSID_MYGCS	199			
SYSID_SW_TYPE	0			
SYSID_THISMAV	1			
TELEM_DELAY	0	seconds	0 10	The amount of time (in seconds) to delay radio telemetry to prevent an Xbee bricking on power up
THR_FAILSAFE	1		0:Disabled,1: Enabled	The throttle failsafe allows you to configure a software failsafe activated by a setting on the throttle input channel
THR_FS_VALUE	950			The PWM level on channel 3 below which throttle failsafe triggers
THR_MAX	100	Percent	0 100	The maximum throttle setting to which the autopilot will apply.
THR_MIN	40	Percent	0 100	The minimum throttle setting to which the autopilot will apply.
THR_SLEWRATE	0	Percent	0 100	maximum percentage change in throttle per second. A setting of 10 means to not change the throttle by more than 10% of the full throttle range in one second
THR_SUPP_MAN	0		0:Disabled,1: Enabled	When throttle is suppressed in auto mode it is normally forced to zero. If you enable this option, then while suppressed it will be manual throttle. This is useful on petrol engines to hold the idle throttle manually while waiting for takeoff
THROTTLE_NUDGE	1		0:Disabled,1: Enabled	When enabled, this uses the throttle input in auto-throttle modes to 'nudge' the throttle to higher or lower values
TRIM_ARSPD_CM	1800	cm/s		Airspeed in cm/s to aim for when airspeed is enabled in auto mode
TRIM_AUTO	0			
TRIM_PITCH_CD	0	centi-Degrees		offset to add to pitch - used for trimming tail draggers
TRIM_THROTTLE	45	Percent	0 100	The target percentage of throttle to apply for normal flight
VOLT_DIVIDER	3.56			

WHEELSTEER_D	0			
WHEELSTEER_I	0			
WHEELSTEER_I MAX?	0			
WHEELSTEER_P	0			
WP_LOITER_RA D	80	Meter s	1 32767	Defines the distance from the waypoint center, the plane will maintain during a loiter
WP_RADIUS	70	Meter s	1 127	Defines the distance from a waypoint, that when crossed indicates the wp has been hit.
XTRK_ANGLE_C D	30 00	centi-Degre es	0 9000	Maximum angle used to correct for track following.
XTRK_GAIN_SC	30		0 2000	The scale between distance off the line and angle to meet the line (in Degrees * 100)
XTRK_MIN_DIST	50	Meter s	0 32767	Minimum distance in meters between waypoints to do crosstrack correction.
XTRK_USE_WIN D	1		0:Disabled,1: Enabled	If enabled, use wind estimation for navigation crosstrack when using a compass for yaw
YW2SRV_D	0.0 01			
YW2SRV_I	0.1			
YW2SRV_IMAX	50 0			
YW2SRV_P	0.5			

Overhead Watch and Loiter (OWL) Advanced Parameter List

This list is the all parameter settings used for well-adjusted autopilot flight of the Sig-Rascal.

AHRS_YAW_P	0.2		0.1 0.4	This controls the weight the compass or GPS has on the heading. A higher value means the heading will track the yaw source (GPS or compass) more rapidly.
ALT_HOLD_FB WCM	0			

ALT_HOLD_RTL	10000	centimeters		Return to launch target altitude
ALT_MIX	1	Percent	0 1	The percent of mixing between gps altitude and baro altitude. 0 = 100% gps, 1 = 100% baro
ALT2PTCH_D	0			
ALT2PTCH_I	0.1			
ALT2PTCH_IMAX	500			
ALT2PTCH_P	0.65			
AMP_PER_VOLT	27.32			
ARSP2PTCH_D	0			
ARSP2PTCH_I	0.1			
ARSP2PTCH_IMAX	500			
ARSP2PTCH_P	0.85			
ARSPD_ENABLE	1		0:Disable,1:Enable	enable airspeed sensor
ARSPD_FBW_MAX	22	m/s	5 50	Airspeed corresponding to maximum throttle in Fly By Wire B mode.
ARSPD_FBW_MIN	6	m/s	5 50	Airspeed corresponding to minimum throttle in Fly By Wire B mode.
ARSPD_OFFSET	2086			Airspeed calibration offset
ARSPD_RATIO	1.994			Airspeed calibration ratio

ARSPD_USE	1		1:Use,0:Don't Use	use airspeed for flight control
BATT_CAPACITY	1760	mAh		Capacity of the battery in mAh when full
BATT_MONITOR	0			
CMD_INDEX	0			
CMD_TOTAL	5			
COMPASS_AUTODEC	1			
COMPASS_DEC	0			
COMPASS_LEARN	1			
COMPASS_OFS_X	-113.393			
COMPASS_OFS_Y	-9.333			
COMPASS_OFS_Z	-111.727			
COMPASS_USE	1			
ELEVON_CH1_REV	0		-1:Disabled,1:Enabled	Reverse elevon channel 1
ELEVON_CH2_REV	0		-1:Disabled,1:Enabled	Reverse elevon channel 2
ELEVON_MIXING	0			
ELEVON_REVERSE	0		0:Disabled,1:Enabled	Reverse elevon mixing

ENRGY2THR_D	0			
ENRGY2THR_I	0			
ENRGY2THR_IMAX	20			
ENRGY2THR_P	0.6			
FENCE_ACTION	0		0:None,1:GuidedMode,2:ReportOnly	What to do on fence breach
FENCE_CHANNEL	0			RC Channel to use to enable geofence. PWM input above 1750 enables the geofence
FENCE_MAXALT	0	meters	0 32767	Maximum altitude allowed before geofence triggers
FENCE_MINALT	0	meters	0 32767	Minimum altitude allowed before geofence triggers
FENCE_TOTAL	0			Number of geofence points currently loaded
FLAP_1_PERCNT	0			
FLAP_1_SPEED	-1			
FLAP_2_PERCNT	0			
FLAP_2_SPEED	-1			
FLTMODE_CH	8			RC Channel to use for flight mode control
FLTMODE1	11		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Loiter,15:Guided	Flight mode for switch position 1 (910 to 1230 and above 2049)
FLTMODE2	11		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Lo	Flight mode for switch position 2 (1231 to 1360)

			iter,15:Guided	
FLTMODE3	10		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Loiter,15:Guided	Flight mode for switch position 3 (1361 to 1490)
FLTMODE4	2		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Loiter,15:Guided	Flight mode for switch position 4 (1491 to 1620)
FLTMODE5	2		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Loiter,15:Guided	Flight mode for switch position 5 (1621 to 1749)
FLTMODE6	0		0:Manual,1:CIRCLE,2:STABILIZE,5:FBWA,6:FBWB,10:Auto,11:RTL,12:Loiter,15:Guided	Flight mode for switch position 6 (1750 to 2049)
FORMAT_VERSION	13			
FS_GCS_ENABL	0		0:Disabled,1:Enabled	Enable ground control station telemetry failsafe. Failsafe will trigger after 20 seconds of no MAVLink heartbeat messages
FS_LONG_ACTN	0		0:None,1:ReturnToLaunch	The action to take on a long (20 second) failsafe event
FS_SHORT_ACTN	0		0:None,1:ReturnToLaunch	The action to take on a short (1 second) failsafe event
GND_ABS_PRESS	99917			
GND_TEMP	25			
HDNG2RLL_D	0.02			
HDNG2RLL_I	0.1			
HDNG2RLL_IMAX	500			

HDNG2RLL_P	0.6			
IMU_PRODUCT_ID	88			
INPUT_VOLTS	4.68			
INVERTEDFLT_CH	0			
KFF_PTCH2THR	0		0 5	Pitch to throttle feed-forward gain.
KFF_PTCHCOMP	0.2		0 1	Adds pitch input to compensate for the loss of lift due to roll control. 0 = 0 %, 1 = 100%
KFF_RDDRMIX	0.5		0 1	The amount of rudder mix to apply during aileron movement 0 = 0 %, 1 = 100%
KFF_THR2PTCH	0		0 5	Throttle to pitch feed-forward gain.
LIM_PITCH_MAX	2000	centi-Degrees	0 9000	The maximum commanded pitch up angle
LIM_PITCH_MIN	-2000	centi-Degrees	-9000 0	The minimum commanded pitch down angle
LIM_ROLL_CD	4500	centi-Degrees	0 9000	The maximum commanded bank angle in either direction
LOG_BITMASK	334			bitmap of log fields to enable
LOG_LASTFILE	0			
MAG_ENABLE	1		0:Disabled,1:Enabled	Setting this to Enabled(1) will enable the compass. Setting this to Disabled(0) will disable the compass
MANUAL_LEVEL	0		0:Disabled,1:Enabled	Setting this to Disabled(0) will enable autolevel on every boot. Setting it to Enabled(1) will do a calibration only when you tell it to
MIN_GNDSPD_CM	0	cm/s		Minimum ground speed in cm/s when under airspeed control

PTCH2SRV_D	0			
PTCH2SRV_I	0.05			
PTCH2SRV_IMAX	500			
PTCH2SRV_P	1			
RC1_DZ	30			dead zone around trim.
RC1_MAX	1834	ms	800 2200	RC maximum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC1_MIN	1274	ms	800 2200	RC minimum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC1_REV	-1		-1:Reversed,1:Normal	Reverse servo operation. Ignored on APM1 unless dip-switches are disabled.
RC1_TRIM	1501	ms	800 2200	RC trim (neutral) PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC2_DZ	30			dead zone around trim.
RC2_MAX	1703	ms	800 2200	RC maximum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC2_MIN	1345	ms	800 2200	RC minimum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC2_REV	-1		-1:Reversed,1:Normal	Reverse servo operation. Ignored on APM1 unless dip-switches are disabled.
RC2_TRIM	1501	ms	800 2200	RC trim (neutral) PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC3_DZ	3			dead zone around trim.
RC3_MAX	2011	ms	800 2200	RC maximum PWM pulse width. Typically 1000 is lower limit, 1500 is

				neutral and 2000 is upper limit.
RC3_MIN	989	ms	800 2200	RC minimum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC3_REV	1		-1:Reversed,1:Normal	Reverse servo operation. Ignored on APM1 unless dip-switches are disabled.
RC3_TRIM	990	ms	800 2200	RC trim (neutral) PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC4_DZ	30			dead zone around trim.
RC4_MAX	1498	ms	800 2200	RC maximum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC4_MIN	1497	ms	800 2200	RC minimum PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC4_REV	1		-1:Reversed,1:Normal	Reverse servo operation. Ignored on APM1 unless dip-switches are disabled.
RC4_TRIM	1498	ms	800 2200	RC trim (neutral) PWM pulse width. Typically 1000 is lower limit, 1500 is neutral and 2000 is upper limit.
RC5_ANGLE_MAX	4500			
RC5_ANGLE_MIN	-4500			
RC5_DZ	0			
RC5_FUNCTION	0		0:Disabled,1:Manual,2:Flap,3:Flap_auto,4:Aileron,5:flaperon,6:mount_pan,7:mount_tilt,8:mount_roll,9:mount_open,10:camera_trigger,11:release,12:mount2_pan,13:mount2_tilt,14:mount2_roll,15:mount2_open,16:DifferentialSpoiler1,17:DifferentialSp	Setting this to Disabled(0) will disable this output, any other value will enable the corresponding function

			oiler2,18:AileronWithInput	
RC5_MAX	1553			
RC5_MIN	1552			
RC5_REV	1			
RC5_TRIM	1553			
RC6_ANGLE_MAX	4500			
RC6_ANGLE_MIN	-4500			
RC6_DZ	0			
RC6_FUNCTION	0		0:Disabled,1:Manual,2:Flap,3:Flap_auto,4:Aileron,5:flaperon,6:mount_pan,7:mount_tilt,8:mount_roll,9:mount_open,10:camera_trigger,11:release,12:mount2_pan,13:mount2_tilt,14:mount2_roll,15:mount2_open,16:DifferentialSpoiler1,17:DifferentialSpoiler2,18:AileronWithInput	Setting this to Disabled(0) will disable this output, any other value will enable the corresponding function
RC6_MAX	1498			
RC6_MIN	1497			
RC6_REV	1			
RC6_TRIM	1498			

RC7_ANGLE_MAX	4500			
RC7_ANGLE_MIN	-4500			
RC7_DZ	0			
RC7_FUNCTION	0		0:Disabled,1:Manual,2:Flap,3:Flap_auto,4:Aileron,5:flaperon,6:mount_pan,7:mount_tilt,8:mount_roll,9:mount_open,10:camera_trigger,11:release,12:mount2_pan,13:mount2_tilt,14:mount2_roll,15:mount2_open,16:DifferentialSpoiler1,17:DifferentialSpoiler2,18:AileronWithInput	Setting this to Disabled(0) will disable this output, any other value will enable the corresponding function
RC7_MAX	1498			
RC7_MIN	1497			
RC7_REV	1			
RC7_TRIM	1498			
RC8_ANGLE_MAX	4500			
RC8_ANGLE_MIN	-4500			
RC8_DZ	0			
RC8_FUNCTION	0		0:Disabled,1:Manual,2:Flap,3:Flap_auto,4:Aileron,5:flaperon,6:mount_pan,7:mount_tilt,8:mount_roll,9:mount_ope	Setting this to Disabled(0) will disable this output, any other value will enable the corresponding function

			n,10:camera_trigger,11:release,12:mount2_pan,13:mount2_tilt,14:mount2_roll,15:mount2_open,16:DifferentialSpoiler1,17:DifferentialSpoiler2,18:AileronWithInput	
RC8_MAX	2015			
RC8_MIN	1246			
RC8_REV	1			
RC8_TRIM	2015			
RLL2SRV_D	0			
RLL2SRV_I	0.12			
RLL2SRV_IMAX	600			
RLL2SRV_P	0.2			
RST_SWITCH_CH	0			RC channel to use to reset to last flight mode after geofence takeover.
SERIAL3_BAUD	57		1:1200,2:2400,4:4800,9:9600,19:19200,38:38400,57:57600,111:111100,115:115200	The baud rate used on the telemetry port
SONAR_ENABLE	0			
SR0_EXT_STAT	2			
SR0_EXTRA1	10			

SR0_EXTRA2	10				
SR0_EXTRA3	2				
SR0_PARAMS	50				
SR0_POSITION	3				
SR0_RAW_CTRL	0				
SR0_RAW_SENS	0				
SR0_RC_CHAN	2				
SR3_EXT_STAT	0				
SR3_EXTRA1	0				
SR3_EXTRA2	0				
SR3_EXTRA3	0				
SR3_PARAMS	0				
SR3_POSITION	0				
SR3_RAW_CTRL	0				
SR3_RAW_SENS	0				
SR3_RC_CHAN	0				

SWITCH_ENABLE	0			
SYS_NUM_RESETS	51			Number of APM board resets
SYSID_MYGCS	255			
SYSID_SW_TYPE	0			
SYSID_THISMAV	99			
THR_FAILSAFE	1		0:Disabled,1:Enabled	The throttle failsafe allows you to configure a software failsafe activated by a setting on the throttle input channel
THR_FS_VALUE	950			The PWM level on channel 3 below which throttle failsafe triggers
THR_MAX	100	Percent	0 100	The maximum throttle setting to which the autopilot will apply.
THR_MIN	0	Percent	0 100	The minimum throttle setting to which the autopilot will apply.
THR_SLEWRATE	0	Percent	0 100	maximum percentage change in throttle per second. A setting of 10 means to not change the throttle by more than 10% of the full throttle range in one second
TRIM_ARSPD_CM	1300	cm/s		Airspeed in cm/s to aim for when airspeed is enabled in auto mode
TRIM_AUTO	0			
TRIM_PITCH_CD	0	centi-Degrees		offset to add to pitch - used for trimming tail draggers
TRIM_THROTTLE	65	Percent	0 100	The target percentage of throttle to apply for normal flight
VOLT_DIVIDER	3.56			

WP_LOITER_RAD	45	Meters	1 32767	Defines the distance from the waypoint center, the plane will maintain during a loiter
WP_RADIUS	30	Meters	1 127	Defines the distance from a waypoint, that when crossed indicates the wp has been hit.
XTRK_ANGLE_CD	3000	centi-Degrees	0 9000	Maximum angle used to correct for track following.
XTRK_GAIN_SC	75		0 2000	The scale between distance off the line and angle to meet the line (in Degrees * 100)
YW2SRV_D	0			
YW2SRV_I	0			
YW2SRV_IMAX	0			
YW2SRV_P	0			

Bibliography

[1] D. Fulghum and L. Dickerson, "UAS Demand is Solid," *Aviation Week & Space Technology*, vol. 174, no. 4, pp. 68-74, 2012.

[2] United States Air Force, "United States Air Force Unmanned Aircraft Systems Flight plan 2009-2047," United States Air Force, Washington DC, 2009.

[3] A. Ryan, M. Zennaro, A. Howell, R. Sengupta and J. K. Hedrick, "An Overview of Emerging Results in Cooperative UAV Control," in *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004.

[4] M. Pachter, J. Hansen, D. Jacques and P. Blue, "Optimal Guidance of a Relay Aircraft to Extend Small Unmanned Aircraft Range," *International Journal of Micro Air Vehicles*, vol. 2, no. 3, pp. 157-180, 2010.

[5] J. P. Boire, "Autonomous Routing of Unmanned Aerial Vehicle(UAV) Relays to Mimic Optimal Trajectories in Real Time," Air Force Institute of Technology, Wright-Patterson AFB, 2011.

[6] M. T. Seibert, A. J. Stryker and C. T. Wellbaum, "System Analysis and Prototyping for Single Operator Management of Multiple Unmanned Aerial Vehicles Operating Beyond Line of Sight," Air Force Institute of Technology, Wright-Patterson AFB , 2010.

[7] C. Youngdong, M. Pachter and D. Jacques, "Optimal Relay UAV Guidance-A New Differentail Game," Air Force Institute of Technology, Wright-Patterson AFB, 2010.

[8] K. Forsberg, H. Mooz and H. Cotterman, Visualizing Project Management: A Model for Business and Technical Success, New York: John Wiley & Sons, Inc., 2000.

[9] United States Air Force, *Air Force Doctrine Document 1*, Washington D.C.: Office of the USAF Chief of Staff, 2003.

[10] United States Department of Defense, "FY 2009-2034 Unmanned System Integrated

Roadmap," Department of Defense, Washington D.C., 2009.

[11] I. Rubin, "Placement of UAVs as Communication Relays Aiding Mobile Ad Hoc Wireless Networks," in *Military Communications Conference*, Los Angeles, 2007.

[12] O. Burdakov, P. Doherty, K. Holmberg and P.-M. Olsson, "Optimal Placement of UV-based Communications Relay Nodes," *Journal of Global Optimization*, vol. 48, no. 4, pp. 551-531, 2010.

[13] B. S. Blanchard and W. J. Fabrycky, Systems Engineering and Analysis, 5th ed., Upper Saddle River: Pearson, 2011.

[14] L. Meier, J. Camacho, B. Godbolt, J. Goppert, L. Heng, M. Lizarraga, L. Meier, K. Rudin, P. Tanskanen and A. Tridgell, "QGroundControl: Ground Control Station for Small Air-Land-Water Autonomous Unmanned Systems," 2010. [Online]. Available: http://qgroundcontrol.org/start. [Accessed 12 August 2012].

[15] M. Oborne, O. Adler, R. Beall, S. Benigno, J. Birkeland, J. Goppert, P. Hickey, J. Hirvinen, H. Hollander, J. Julio, S. Kelly, M. Levine, R. Mackay, P. Mather, J. Munoz, R. Navoni and S. Nelson, "ArduPlane," Google, 1 July 2012. [Online]. Available: http://code.google.com/p/ardupilot-mega/wiki/home?tm=6. [Accessed 25 January 2013].

[16] S. Songer, "Aerial Networking for the Implementation of Cooperative Control on SUAS," Air Force Institute of Technology, Wright-Patterson AFB, 2013.

[17] FrSky, "Products Show: Sensor Hub and Sensors," FrSky, 2010. [Online]. Available: http://www.frsky-rc.com/ShowProducts.asp?id=92. [Accessed 9 January 2013].

[18] D. Conger, J. Stew and G. Hattenberger, "DIY Drones: I Need A History Lesson. How Did We Get Here?," NING| GLAMSOCIAL, 18 August 2012. [Online]. Available: http://www.diydrones.com/forum/topics/i-need-a-history-lesson-how-did-we-get-here. [Accessed 29 January 2013].

Acronym List

1.	AFIT	Air Force Institute of Technology
2.	APM	Ardupilot Mega
3.	COTS	Commercial-Off-The-Shelf
4.	GOTS	Government-Off-The-Shelf
5.	GPS	Global Positioning System
6.	ISR	Intelligence Surveillance and Reconnaissance
7.	MAV	Micro Aerial Vehicle
8.	MAVLink	Micro Aerial Vehicle Link
9.	OV	Operational View
10.	OWL	Overhead Watch and Loiter
11.	RC	Radio Control
12.	SIL	Software In the Loop
13.	SUAS	Small Unmanned Aerial System
14.	UAV	Unmanned Aerial Vehicle

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
21-03-2013	Master's Thesis	March 2012 – March 2013

TITLE AND SUBTITLE	
Development of Autonomous Optimal Cooperative Control in Relay Rover Configured Small Unmanned Aerial Systems	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR	**5d. PROJECT NUMBER**
Shuck, Timothy, J., 1st Lieutenant, USAF	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	AFIT-ENV-13-M-27

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
AFRL/RQQA Mr. Derek Kingston 2210 8th St, Wright-Patterson AFB, OH Phone: (937) 255-6301 Email: derek.kingston@wpafb.af.mil	AFRL/RQQA **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This thesis documents the research effort to develop, integrate and implement the system hardware and the software necessary to validate the Air Force Institute of Technology's theoretical advances in small unmanned aerial systems (SUAS) cooperative control. The end state objective of the research effort was to flight test an autonomous control algorithm on a communication relay unmanned aerial vehicle (UAV) that was actively relaying data to and from a rover UAV. The relay UAV is one part of a SUAS designed to utilize cooperative control to extend the effective line-of-sight operating range for a rover UAV. An algorithm is integrated into ground control software that takes telemetry data (the current position of the ground station, rover UAV, and relay UAV) to determine where to navigate the relay aircraft for optimal communication signal strength. The ground station operator flies the rover aircraft in the extended line-of-sight operational envelope just as she/he would in the normal line-of-sight operations. The relay UAV is autonomously routed to the optimal communications relay position.
The research yielded a SUAS based on the Ardupilot Mega 2.0. Flight testing demonstrated the SUAS's ability to generate the correct navigation data autonomously; however, the navigation data was not successfully activated as current waypoints on the relay UAV's autopilot. Software in the loop testing was utilized to verify a solution to activate the navigation data but flight testing was not conducted to verify the simulation results.

15. SUBJECT TERMS
SUAS, Multi-Aircraft Control, Cooperative Control, UAV

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					David Jacques, AFIT/ENV, Research Advisor
a. REPORT	b. ABSTRACT	c. THIS PAGE	**UU**	118	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 3329 (david.jacques@afit.edu)
U	U	U			

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18